Calhoun: The NPS Institutional Archive

DSpace Repository

Theses and Dissertations | 1. Thesis and Dissertation Collection, all items

1999-06

Simulation and modeling of a soft grounding system for an AUV

Beyazay, Bahadir.

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/13492

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

SIMULATION AND MODELING OF A SOFT GROUNDING SYSTEM FOR AN AUV

by

Bahadir Beyazay

June 1999

Thesis Advisor: Anthony J. Healey

Approved for public release; distribution is unlimited

19990817 102

		Form Approved
REPORT DOCUMENTATION PAGE		**OMB No. 0704-0188**

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(leave bank)*	2. REPORT DATE June 1999	3. REPORT TYPE AND DATES COVERED Master's Thesis

4. TITLE AND SUBTITLE Simulation and Modeling of a Soft Grounding System for an AUV	5.FUNDING NUMBERS
6. AUTHOR(S) Bahadir Beyazay	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000	8.PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.	12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

Energy storage is very limited in AUV's. To assist with energy management, data gathering missions have been proposed where the vehicle should sit on the bottom and gather acoustic/video/chemical data over extended periods of time. In this grounding senario while thrusters may be used, they are less desirable because of their high energy consuption and restricted use close to the ocean floor. The purpose of this work is to study a low cost, simple soft grounding capability for a submersible vehicle using controllable ballast. The ballast system based on the NPS Phoenix AUV is designed to control weight addition into or out of two ballast tanks. The developed control law adjusts the pump flow rate keeping the pitch angle and depth rate within the limits. Results for a soft grounding operation have been obtained using simulation.

14. SUBJECT TERMS Sliding Mode Control, Linear Quadratic Regulator, Autonomous Underwater Vehicles, Flight and Ballast Control Systems, Bottom Stability	5.NUMBER OF PAGES 95
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	9. SECURITY CLASSIFICATION OF BSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18298102

SIMULATION AND MODELING OF A SOFT GROUNDING SYSTEM FOR AN AUV

Bahadir Beyazay
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 1993

Submitted in partial fulfillment of the
requirements for the degree of

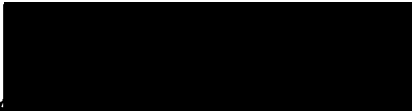## MASTER OF SCIENCE IN MECHANICAL ENGINEERING
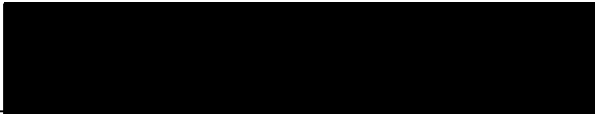
from the

NAVAL POSTGRADUATE SCHOOL
June 1999

Author: _____

Bahadir Beyazay

Approved by:_____

Anthony J. Healey, Professor of Mechanical Engineering,
Thesis Advisor

Approved by: _____

Terry R. McNelley, Chair, Department of Mechanical Engineering

ABSTRACT

Energy storage is very limited in AUV's. To assist with energy management, data gathering missions have been proposed where the vehicle should sit on the bottom and gather acoustic/video/chemical data over extended periods of time. In this grounding senario while thrusters may be used, they are less desirable because of their high energy consuption and restricted use close to the ocean floor. The purpose of this work is to study a low cost, simple soft grounding capability for a submersible vehicle using controllable ballast. The ballast system based on the NPS Phoenix AUV is designed to control weight addition into or out of two ballast tanks. The developed control law adjusts the pump flow rate keeping the pitch angle and depth rate within the limits. Results for a soft grounding operation have been obtained using simulation.

TABLE OF CONTENTS

LIST OF FIGURES

ACKNOWLEDGEMENTS

I.   INTRODUCTION

A.   INTRODUCTION

Energy storage is very limited in AUV's. To assist with energy management, data gathering missions have been proposed where the vehicle should sit on the bottom and gather acoustic/video/chemical data over extended periods of time. In this grounding senario, thrusters may be used. However, there are two disadvantages for this method: high energy consumption and restricted use close to the ocean bottom. The motivation for this thesis is to study a low cost, simple soft grounding capability for a submersible vehicle using controllable ballast. For simplicity, water ballast is considered. The design of the control system is based on the NPS Phoenix AUV. The ballast system is designed to control the weight addition into or out of the two ballast tanks where instead of a bang-bang or a fuzzy logic controller which actually turns on and turns off the pumps , the developed control law controls the pump flow rate, keep the pitch angle and depth rate within limits.

B.   BACKGROUND

Ballast control of vehicles is not a new subject and we can find many examples beginning in the 1900's, the non-rigid airships are very good examples of ballast control. One of the most important elements of a non-rigid airship is the ballonet-system. A ballonet as seen in Figure 1-1 is an airbag (one or two of them) inside the envelope, which is provided with air from a blower or directly from the engine unit. The air could be removed from the ballonet through the valves. If the airship has a front and aft ballonet then the height of the airship can be steered. For example, if the aft ballonet is filled with more air, then the airship will become heavier in the rear part of the envelope and the ship will incline increasing the altitude of the ship by using the engines.As Figure 1-2 depicted, the airship can also be trimed through a front and aft ballonet [Ref. 1].

1

For most of the maneuvering underwater vehicles, the depth control is normally provided by hydroplanes. As an example, consider the NPS Phoenix AUV. It has two sets of control surfaces, namely the bow and the stern planes. At low speed, the control surfaces cannot provide enough control authority and the ballast control problem is very complex due to nonlinear, time-varying, uncertain hydrodynamics. There are some designs that used bang-bang control system [Ref. 2]. The ARPA's Unmanned Undersea Vehicle (UUV) employs a fuzzy logic ballast controller which is comparable with the performance that can be obtained from standard control techniques, but does not require traditional linear or nonlinear design methods.
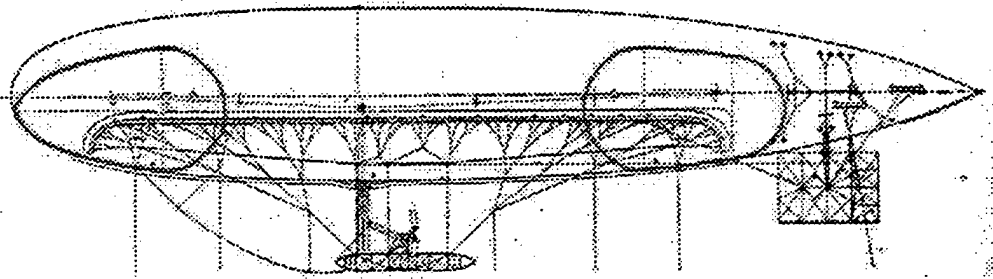
Figure 1-1. Sectional elevation of the Parseval-Airship "PL VI", 1910.
(Marked areas are the ballonets inside the envelope) From [Ref. 1]
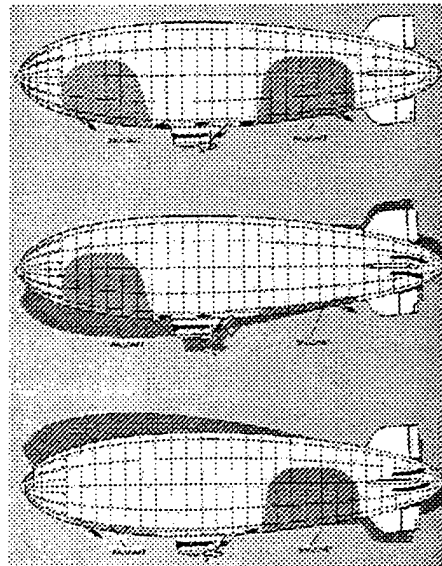
Figure 1-2. Elevator controling via front and aft ballonet, (newer type).
From [Ref. 1]

2

In another fuzzy logic control model, a 15 state Kalman filter was developed to provide estimates of the motion variables and the applied lift and torque acting on the UUV. The control law decides between three possible control actions; pump water in both tanks, pump water out of both tanks and turn both pumps off. The fuzzy input state space is composed of depth error and depth rate, and each is divided into partitions. The fuzzy controller interpolates between the partitions allowing the control to vary smoothly as the states move from one partition to another. These movements of states were provided by on and off of ballast pumps [Ref. 3].

C.   THESIS MOTIVATION AND GOALS

The thesis will outline the development of a depth controller using sliding mode control techniques for a neutrally buoyant vehicle. The sliding mode controller is designed on the basis of the simplified four degrees of freedom vertical plane equations of motion.

A linear quadratic regulator (LQR) proportional approach is then utilized for the design of the ballast controller, which produces flow rate commands, allowing the vehicle to have a soft grounding behavior.

These two controllers used a logic based depth regulator to provide realistic simulation of the vehicle's flight and grounding capabilities for a single mission.

D.   THESIS ORGANIZATION

In Chapter II, the equations of motion in six degrees of fredom are reviewed simplified for vertical plane case [Ref. 4], which provides the basis for the depth control design. Chapter III discusses the design of a flight controller with sliding mode techniques, and the development of the grounding controller using different methods of ballast control. Chapter IV outlines the methods and functions used for the simulating the designed systems. Chapter V discusses the results of the simulation studies which were conducted by using parameters of NPS Phoenix AUV. The hardware components like sensors, pumps, and tanks needed for ballast control explained in Chapter VI. The conclusions and recommendations can be found in Chapter VII.

II. VEHICLE MODELING AND EQUATIONS OF MOTION

A. INTRODUCTION

For simulation of maneuvering and motion control of the vehicle, it is assumed that [Ref.4] :

1. The vehicle behaves as a rigid body

2. The earth's rotation is neglected as far as acceleration components of the vehicle's center of mass is concerned.

3. The primary forces that act on the vehicle have inertial, gravitational origins and hydrostatic, propulsion, thruster, and hydrodynamic forces from lift and drag.

B. DEVELOPMENT OF THE SIX DEGREES OF FREEDOM NON-LINEAR EQUATIONS OF MOTION FOR A MARINE VEHICLE

The following work concerned with development of the six degrees of freedom non-linear equations of motion for a marine vehicle includes the equations described by Healey [Ref. 4]. A vector $\vec{x}$ of vehicle body frame velocities and a vector $\vec{z}$ of global positions was defined as:

$$\vec{x} = \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix} \quad \text{and} \quad \vec{z} = \begin{bmatrix} X \\ Y \\ Z \\ \phi \\ \theta \\ \psi \end{bmatrix} \tag{2.1}$$

then considering M as a 6 x 6 mass matrix including translational and rotational inertial elements, the equations of motion can be written in the following vector form,

$$M\dot{\vec{x}} + f(x) + \vec{F}_g(z) = \vec{F}_h \tag{2.2}$$

and,

5

$$\dot{z} + g(x, z) = 0 \qquad (2.3)$$

With suitable knowledge of the excitation force and moment loads, a solution for the vehicle's dynamics can be obtained. A more detailed insight into the development of these differential equations, in first order form given by the foregoing analysis shows,

$$m\{\dot{\vec{v}} + \dot{\vec{w}} \times \vec{\rho}_G\} + m(\vec{w} \times \vec{w} \times \vec{\rho}_G + \vec{w} \times v) + f_G(z) = \begin{bmatrix} X_f \\ Y_f \\ Z_f \end{bmatrix}, \qquad (2.4)$$

and,

$$I_o \dot{\vec{w}} + m\{\vec{\rho}_G \times \dot{\vec{v}}\} + \vec{w} \times (I_o \vec{w}) + m\{\vec{\rho}_G \times \vec{w} \times \vec{v}\} + \vec{m}_g(z) = \begin{bmatrix} K_f \\ M_f \\ N_f \end{bmatrix} \qquad (2.5)$$

It helps here to define the cross product coefficient matrix so that,

$$\dot{\vec{w}} \times \vec{\rho}_G = -\vec{\rho}_G \times w = -[cros(\vec{\rho}_G)]\dot{\vec{w}} \qquad (2.6)$$

where,

$$[cros(\vec{\rho}_G)] = \begin{bmatrix} 0 & z_G & -y_G \\ -z_G & 0 & x_G \\ y_G & -x_G & 0 \end{bmatrix} \qquad (2.7)$$

The next step is to collect all inertial terms into a 6 x 6 mass matrix including the inertia cross coupling effects,

$$M = \begin{bmatrix} \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix} & m\begin{bmatrix} 0 & z_G & -y_G \\ -z_G & 0 & x_G \\ y_G & -x_G & 0 \end{bmatrix} \\ m\begin{bmatrix} 0 & -z_G & y_G \\ z_G & 0 & -x_G \\ -y_G & x_G & 0 \end{bmatrix} & \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \end{bmatrix} \qquad (2.8)$$

The remaining terms on the left hand side arising from centripetal and coriolis accelerations become,

6

$$f(x)=\begin{bmatrix} m(w.\times w\times .\rho_G + w\times .v) \\ w\times .(I_o w) + m\{\rho_G \times w\times .v\} \end{bmatrix} \qquad (2.9)$$

The double vector cross products are nonlinear in the primary velocity variables and hence the need for the nonlinear functional, $f(\bullet)$. The reader can perform the indicated manipulations to express individual equations within the set.It may also help if the screw symetric matrix $S(\bullet)$ is used in place of the vector cross product [Ref. 5].This yields,

$$f(x)=\begin{bmatrix} m(S(w).S(w).\rho_G + S(w).v) \\ S(w).(I_o w) + m\{S(\rho_G)S(w).v\} \end{bmatrix} \qquad (2.10)$$

The screw symetric matrix involving the vector cross product of ω with other vector is

$$S(\omega) = \begin{bmatrix} 0 & -r & +q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \qquad (2.11)$$

The components of the sum of all external forces and moments acting on the vehicle body are separated in the above analysis into six components each acting along the vehicle body fixed coordinate axes and form the total vector of forces and moments as

$$F(t)=[X_f(t),Y_f(t),Z_f(t),K_f(t),M_f(t),N_f(t)]' \qquad (2.12)$$

where the vector components in order refer to the surge, sway, heave forces, and the roll, pitch, and yaw moments respectively.The expressions for X_f etc. are found as the sum of all external forces acting on the vehicle.

In the component form, these external forces can be defined as follows,

$$m[\dot{u} + qw - rv - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] = X$$
$$m[\dot{v} + ru - pw - y_G(r^2 + p^2) + z_G(qr - \dot{p}) + x_G(qp + \dot{r})] = Y \qquad (2.13)$$
$$m[\dot{w} + pv - qu - z_G(p^2 + q^2) + x_G(rp - \dot{q}) + y_G(rq + \dot{p})] = Z$$

where

$$\vec{F} = X\hat{i} + Y\hat{j} + Z\hat{k} \qquad (2.14)$$

7

If $\vec{\overline{I}}$ is the mass matrix of inertia of the body, in component form,

$$\vec{H} = H_x \hat{i} + H_y \hat{j} + H_z \hat{k} \qquad (2.15)$$

and,

$$
\begin{aligned}
H_x &= I_x p - I_{xy} q - I_{xz} r \\
H_y &= -I_{yx} p + I_y q - I_{yz} r \\
H_z &= -I_{zx} p - I_{zy} q + I_z r
\end{aligned} \qquad (2.16)
$$

If the external moment, $\vec{M}$, is denoted as,

$$\vec{M} = K\hat{i} + M\hat{j} + N\hat{k} \qquad (2.17)$$

and in component form,

$$I_x \dot{p} + (I_z - I_y)qr + I_{xy}(pr - \dot{q}) - I_{yz}(q^2 - r^2) - I_{xz}(pq + \dot{r})$$

$$+ m[y_G(\dot{w} - uq + \upsilon p) - z_G(\dot{\upsilon} + ur - wp)] = K \quad , \qquad (2.18)$$

$$I_y \dot{q} + (I_x - I_z)pr + I_{xy}(qr + \dot{p}) + I_{yz}(pq - \dot{r}) + I_{xz}(p^2 - r^2)$$

$$- m[x_G(\dot{w} - uq + \upsilon p) - z_G(\dot{\upsilon} - \upsilon r + wq)] = M \quad , \qquad (2.19)$$

$$I_z \dot{r} + (I_y - I_x)pq - I_{xy}(p^2 - q^2) - I_{yz}(pr + \dot{q}) + I_{xz}(qr - \dot{p})$$

$$+ m[x_G(\dot{\upsilon} + ur - wp) - y_G(\dot{u} - \upsilon r + wq)] = N \qquad (2.20)$$

where,

I_x	: vehicle mass moment of inertia around the x-axis
(K,M,N)	: roll, pitch, and yaw moments, respectively
(p,q,r)	: roll, pitch, and yaw rates, respectively
(u, υ, w)	: surge, sway, and heave velocities, respectively
x	: body fixed axis, positive forward
x_G	: longitutional position of center of gravity
(X,Y,Z)	: surge, sway, and heave forces, respectively

8

y   : body fixed axis, positive starboard

y_G : athwartship location of center of gravity

z   : body fixed axis, positive down

z_G : vertical position of center of gravity

(ϕ, θ, ψ) : roll, pitch, and yaw Euler angles, respectively

These equations are the three rotational equations of motion, and together with the translational equations, are the six degrees of freedom equations of motion of a rigid body expressed in a coordinate system moving with the body, fixed at the body's geometric center [Ref. 4].

C.   EXTERNAL FORCES AND MOMENTS

The right-hand sides of Equations 2-13 and 2-20 may be expanded to include the sum of individual component forces and moments arising from hydrostatic and hydrodynamic sources and external forces arising from control surface deflections and propeller thrust [Ref 6]. These equations are expressed as follows:

$$m[\dot{u} - \upsilon r + wq - x_G(q^2 + r^2) + y_G(pq - \dot{r}) + z_G(pr + \dot{q})] = X_H + X_W + X_C$$
$$m[\dot{\upsilon} + ur - wq + x_G(pq + \dot{r}) - y_G(p^2 + r^2) + z_G(qr - \dot{p})] = Y_H + Y_W + Y_C \qquad (2.21)$$
$$m[\dot{w} - uq + \upsilon p + x_G(pr - \dot{q}) + y_G(qr + \dot{p}) - z_G(p^2 + q^2)] = Z_H + Z_W + Z_C$$

$$I_x\dot{p} + (I_z - I_y)qr + I_{xy}(pr - \dot{q}) - I_{yz}(q^2 - r^2) - I_{xz}(pq + \dot{r}) \qquad (2.22)$$

$$+ m[y_G(\dot{w} - uq + \upsilon p) - z_G(\dot{\upsilon} + ur - wp)] = K_H + K_W + K_C$$

$$I_y\dot{q} + (I_x - I_z)pr - I_{xy}(qr + \dot{p}) - I_{yz}(pq - \dot{r}) - I_{xz}(p^2 - r^2) \qquad (2.23)$$

$$+ m[x_G(\dot{w} - uq + \upsilon p) - z_G(\dot{u} - \upsilon r + wq)] = M_H + M_W + M_C$$

$$I_z\dot{r} + (I_y - I_x)pq - I_{xy}(p^2 - q^2) - I_{yz}(pr + \dot{q}) + I_{xz}(qr - \dot{p}) \qquad (2.24)$$

$$+ m[x_G(\dot{\upsilon} + ur - wp) - y_G(\dot{u} - \upsilon r - wq)] = N_H + N_W + N_C$$

9

Hydrostatic restoring forces and moments are due to the vehicle weight W and buoyancy B. The net buoyancy force in the inertia system is (W-B) in the positive Z-direction (downwards). Therefore, in the ship fixed system,

$$X_W = -(W - B)\sin\theta$$
$$Y_w = (W - B)\cos\theta\sin\phi \qquad (2.25)$$
$$Z_W = (W - B)\cos\theta\cos\phi$$

The moments due to W and B in the component form,

$$K_W = (y_G W - y_B B)\cos\theta\cos\phi - (z_G W - z_B B)\cos\theta\sin\phi$$
$$M_W = -(x_G W - x_B B)\cos\theta\cos\phi - (z_G W - z_B B)\sin\theta \qquad (2.26)$$
$$N_W = (x_G W - x_B B)\cos\theta\sin\phi + (y_G W - y_B B)\sin\theta$$

Forces and moments due to control surface deflections are reflected as added drag in surge, while in sway, heave, pitch and yaw they are directly proportional to control surface deflection [Ref. 6],

$$X_C = (X_{q\delta_s}\delta_s + X_{q\delta_B}\delta_b)q + X_{r\delta_r}r\delta_r + X_{\upsilon\delta_r}\upsilon\delta_r + (X_{\omega\delta_s}\delta_s + X_{\omega\delta_b}\delta_b)w \qquad (2.27)$$
$$+ X_{\delta_s\delta_s}\delta_s{}^2 + X_{\delta_b\delta_b}\delta_b{}^2 + X_{\delta_r\delta_r}\delta_r{}^2 + X_p(n)$$

$$Y_C = Y_{\delta_r}\delta_r + Y_p(n) \qquad (2.28)$$

$$Z_C = Z_{\delta_S}\delta_s + Z_{\delta_b}\delta_b \qquad (2.29)$$

$$K_C = K_P(n) \qquad (2.30)$$

$$M_C = M_{\delta_s}\delta_s + M_{\delta_b}\delta_b\,\delta \qquad (2.31)$$

$$N_C = N_{\delta_r}\delta_r + M_P(n) \qquad (2.32)$$

where,

δ_r = rudder deflection

δ_s = stern plane deflection

δ_b = bow plane deflection

The terms with subscript P represents the forces and moments generated by the propeller revolutions, n. The most significant of these forces is the propeller thrust, $X_P(n)$, while Y_P, K_P, and N_P represent small asymmetry effects generated by the propeller. These three terms would be zero for the vehicle equipped with two symmetrically loaded counter-rotating propellers.

The hydrodynamic forces and moments are expressed as polynomial functions of the translational and rotational velocities of the vehicle with respect to the water by using a constant coefficient model,

$$X_H = X_{pp}p^2 + X_{qq}q^2 + X_{rr}r^2 + X_{pr}pr + X_{\dot{u}}\dot{u} + X_{wq}wq + X_{\upsilon p}\upsilon p + X_{\upsilon r}\upsilon r \tag{2.33}$$

$$+ X_{\upsilon\upsilon}\upsilon^2 + X_{\omega\omega}w^2 + R(u)$$

$$Y_H = Y_{\dot{p}}\dot{p} + Y_{\dot{r}}\dot{r} + Y_{pq}pq + Y_{qr}qr + Y_{\dot{\upsilon}}\dot{\upsilon} + Y_p p + Y_r r + Y_{\upsilon q}\upsilon q + Y_{\omega p}wp + Y_{wr}wr \tag{2.34}$$

$$+ Y_{\upsilon}\upsilon + Y_{\upsilon w}\upsilon w$$

$$- \frac{1}{2}\rho\int\left[C_{D_Y}h(x)(\upsilon+xr)^2 + C_{D_Z}b(x)(w-xq)^2\right]\frac{(\upsilon+xr)}{U_{cf}(x)}dx$$

$$Z_H = Z_{\dot{q}}\dot{q} + Z_{pp}p^2 + Z_{pr}pr + Z_{rr}r^2 + Z_{\dot{w}}\dot{w} + Z_q q + Z_{\upsilon p}\upsilon p + Z_{\upsilon r}\upsilon r + Z_w w + Z_{\upsilon\upsilon}\upsilon^2 \tag{2.35}$$

$$- \frac{1}{2}\rho\int\left[C_{D_Y}h(x)(\upsilon+xr)^2 + C_{D_Z}b(x)(w-xq)^2\right]\frac{(w-xq)}{U_{cf}(x)}dx$$

$$K_H = K_{\dot{p}}\dot{p} + K_{\dot{r}}\dot{r} + K_{pq}pq + K_{qr}qr + K_{\dot{\upsilon}}\dot{\upsilon} + K_p p + K_r r + K_{\upsilon q}\upsilon q + K_{wp}wp + K_{wr}wr \tag{2.36}$$

$$+ K_{\upsilon}\upsilon + K_{\upsilon w}\upsilon w$$

$$M_H = M_{\dot{q}}\dot{q} + M_{pp}p^2 + M_{pr}pr + M_{rr}r^2 + M_{\dot{w}}\dot{w} + M_q q + M_{\upsilon p}\upsilon p + M_{\upsilon r}\upsilon r + M_w w + M_{\upsilon\upsilon}\upsilon^2$$

$$+ \frac{1}{2}\rho\int\left[C_{D_Y}h(x)(\upsilon+xr)^2 + C_{D_Z}b(x)(w-xq)^2\right]\frac{(w-xq)}{U_{cf}(x)}xdx \tag{2.37}$$

$$N_H = N_{\dot{p}}\dot{p} + N_{\dot{r}}\dot{r} + N_{pq}pq + N_{qr}qr + N_{\dot{\upsilon}}\dot{\upsilon} + N_p p + N_r r + N_{\upsilon q}\upsilon q + N_{wp}wp + N_{wr}wr \tag{2.38}$$

$$+ N_\upsilon \upsilon + N_{\upsilon w} \upsilon w$$

$$-\frac{1}{2}\rho \int \left[C_{D_Y} h(x)(\upsilon + xr)^2 + C_{D_Z} b(x)(w - xq)^2 \right] \frac{(\upsilon + xr)}{U_{cf}(x)} x dx$$

where R(u) represents the vehicle's resistance curve, which is negative since positive direction is ahead. The cross flow integral terms are integrated over the length of the body and they model quadratic drag forces. The cross flow velocity U_{cf} is,

$$U_{cf} = \left[(\upsilon + xr)^2 + (w - xq)^2 \right]^{1/2} \tag{2.39}$$

D. EQUATIONS OF MOTION IN VERTICAL PLANE

We will deal with only vertical plane variables; i.e., heave, pitch, and surge. The vertical plane stability analysis involves heave and pitch motions. However, the surge equation couples into pitch and heave through the metacentric height z_G. This is a dynamic coupling, and could be eliminated by redefining hydrodynamic coefficients with respect to the ship's center of gravity instead of its geometric center.

Restricting the motions of the vehicle to the vertical (dive) plane, the only significant motions that must be incorporated to model the vehicle in the dive plane are, the surge velocity (u), the heave velocity (w), the pitch velocity (q), the pitch angle (θ) and the global depth position (z). This restriction simplifies the previously developed equations to a system of four non-linear equations of motion, which are,

$$\dot{\theta} = q \tag{2.40}$$

$$(m - Z_{\dot{w}})\dot{w} + (-mx_G - Z_{\dot{q}})\dot{q} = (m + Z_q)Uq + mz_G q^2 + Z_w Uw + (W - B)\cos\theta$$

$$-\frac{1}{2}\rho \int_{tail}^{nose} C_D b(x) \frac{(w - xq)^3}{|w - xq|} dx + U^2 (Z_{\delta_s} + \alpha Z_{\delta_b})\delta_s \tag{2.41}$$

12

$$(-mx_G - M_{\dot{w}})\dot{w} + (I_y - M_{\dot{q}})\dot{q} = (M_q - mx_G)Uq - mz_G wq + M_w Uw + U^2(M_{\delta_s} + \alpha M_{\delta_b})\delta_s$$

$$-\frac{1}{2}\rho \int_{tail}^{nose} C_D b(x)\frac{(w-xq)^3}{|w-xq|}xdx - (x_G W - x_B B)\cos\theta - (z_G W - z_B B)\sin\theta \qquad (2.42)$$

$$\dot{z} = -U\sin\theta + w\cos\theta \qquad (2.43)$$

These equations can be linearized for a level flight path when the dive plane angle is zero, $\delta_o=0$. By setting all the time derivatives to zero and neglecting for the moment the hydrodynamic drag terms, the following are obtained :

$$Z_w Uw + (W - B)\cos\theta = 0 \qquad (2.44)$$

$$M_w Uw - (x_G W - x_B B)\cos\theta - (z_G W - z_B B)\sin\theta = 0 \qquad (2.45)$$

$$q = 0 \qquad (2.46)$$

$$-U\sin\theta + w\cos\theta = 0 \qquad (2.47)$$

If the vehicle is neutrally buoyant :

$$\begin{aligned} x_G &= x_B \\ W &= B \end{aligned} \qquad (2.48)$$

Then the Equations 2.40-2.43 are linearized as :

$$\dot{\theta} = q \qquad (2.49)$$

$$(m - Z_{\dot{w}})\dot{w} + (-mx_G - Z_{\dot{q}})\dot{q} = (m + Z_q)Uq + Z_w Uw + U^2 Z_\delta \delta_s \qquad (2.50)$$

$$(-mx_G - M_{\dot{w}})\dot{w} + (I_y - M_{\dot{q}})\dot{q} = (M_q - mx_G)Uq + M_w Uw - (z_G - z_B)W\theta + U^2 M_\delta \delta_s \qquad (2.51)$$

$$\dot{z} = -U\theta + w \qquad (2.52)$$

Both Z_δ and M_δ are a linear combination of the respective stern and bow hydrodynamic control surface coefficients and the respective input value of δ. This makes the system of equations as a multiple input system. To reduce this system into a single input system, the linear combination of control inputs will be modified into the following form,

13

$$Z_\delta = Z_{\delta_s} + \alpha Z_{\delta_b} \quad \text{and} \quad M_\delta = M_{\delta_s} + \alpha M_{\delta_b} \tag{2.53}$$

This will allow a single input δ to control both stern planes and bow planes, and will cause the bow planes to be slaved to the stern planes. This technique is known as dual control. The value of α will range from -1 to 1. The selection of the value of α will allow the planes to operate as desired for the particular maneuvering condition, i.e., $\alpha = 0$ for no bow plane control, $\alpha = -1$ for bow plane and stern plane control opposed to each other, resulting the maximum pitch moment, and $\alpha = 1$ for bow and stern plane control in the same direction, resulting the maximum heave force.

These equations can be shown in matrix format as follows,

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (m - Z_{\dot{w}}) & -(mx_G + Z_{\dot{q}}) & 0 \\ 0 & -(mx_G + M_{\dot{w}}) & (I_y - M_{\dot{q}}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta} \\ \dot{w} \\ \dot{q} \\ \dot{z} \end{bmatrix} =$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & Z_w U & (Z_q + m)U & 0 \\ -(z_G - z_B)W & M_w U & (M_q - mx_G)U & 0 \\ -U & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ w \\ q \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ U^2 Z_\delta \\ U^2 M_\delta \\ 0 \end{bmatrix} \delta \tag{2.54}$$

$$\begin{bmatrix} \dot{\theta} \\ \dot{w} \\ \dot{q} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ a_{21} z_{GB} & a_{22} U & a_{23} U & 0 \\ a_{31} z_{GB} & a_{32} U & a_{33} U & 0 \\ -U & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \theta \\ w \\ q \\ z \end{bmatrix} + \begin{bmatrix} 0 \\ b_1 U^2 \\ b_2 U^2 \\ 0 \end{bmatrix} \delta \tag{2.55}$$

where :

$$Dv = (m - Z_{\dot{w}})(I_y - M_{\dot{q}}) - (mx_G + Z_{\dot{q}})(mx_G + M_{\dot{w}}) \tag{2.56}$$

$$z_{GB} = z_G - z_B \tag{2.57}$$

$$a_{21} = -\frac{(mx_G + Z_{\dot{q}})W}{Dv} \tag{2.58}$$

14

$$a_{22} = \frac{(I_y - M_{\dot{q}})Z_w + (mx_G + Z_{\dot{q}})M_w}{Dv} \tag{2.59}$$

$$a_{23} = \frac{(I_y - M_{\dot{q}})(m + Z_q) + (mx_G + Z_{\dot{q}})(M_q - mx_G)}{Dv} \tag{2.60}$$

$$a_{31} = \frac{-(m - Z_{\dot{w}})W}{Dv} \tag{2.61}$$

$$a_{32} = \frac{(mx_G + M_{\dot{w}})Z_w + (m - Z_{\dot{w}})M_w}{Dv} \tag{2.62}$$

For the case considered during this work, the vehicle has also two ballast tanks which were designed to be used during the grounding. These ballast tanks can be seen in Figure 2-1.
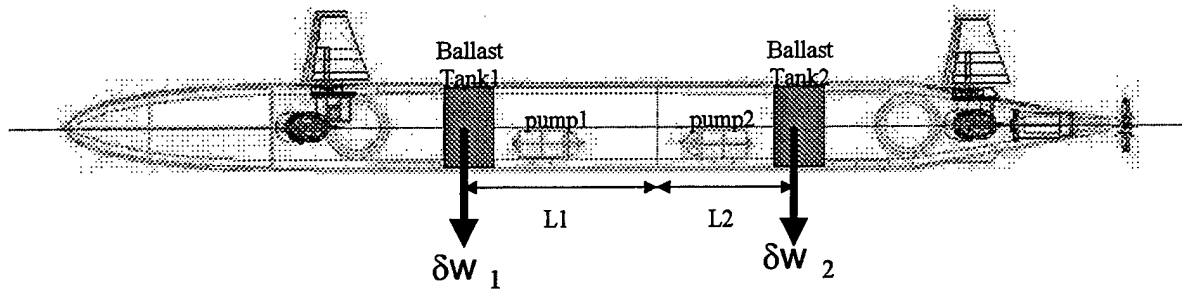
Figure 2-1. The Location of Ballast Tanks in the AUV

The new forces are δw_1 and δw_2 and since the ballast tanks are not located in the same distance from the center of gravity of the vehicle there will be also two moments, $L_1\delta w_1$ and $L_2\delta w_2$. There will be also small change in moment of inertia. So all these changes can be listed as;

$$W = W_o + \delta w_1 + \delta w_2 \tag{2.63}$$

$$m = \frac{(W_o + \delta w_1 + \delta w_2)}{g} \tag{2.64}$$

$$I_y = I_{y_o} + L_1^2(\delta w_1 / g) + L_2^2(\delta w_2 / g) \tag{2.65}$$

And the new equations of motion become,

$$\dot{\theta} = q \tag{2.66}$$

15

$$(m - Z_{\dot{w}})\dot{w} - (mx_G + Z_{\dot{q}})\dot{q} = (m + Z_q)uq + mz_Gq^2 + Z_wuw + (W - B)\cos\theta \qquad (2.67)$$

$$-\frac{1}{2}\rho\int C_D b(x)\frac{(w - xq)^3}{|w - xq|} + u^2 Z_d\delta_s$$

$$(-mx_G - M_{\dot{w}})\dot{w} + (I_y - M_{\dot{q}})\dot{q} = (M_q - mx_G)uq - mz_Gwq + M_wuw - (x_GW - x_BB)\cos\theta$$

$$-\frac{1}{2}\rho\int C_D b(x)\frac{(w - xq)^3}{|w - xq|}xdx - (z_GW - z_BB)\sin\theta + u^2 M_d\delta_s + (L_2\delta w_2 - L_1\delta w_1)\cos\theta$$

$$(2.68)$$

$$\dot{z} = -u\sin\theta + w\cos\theta \qquad (2.69)$$

After linearization,

$$\dot{\theta} = q \qquad (2.70)$$

$$(m - Z_{\dot{w}})\dot{w} - (mx_G + Z_{\dot{q}})\dot{q} = (m + Z_q)uq + Z_wuw + (W - B) - \frac{1}{2}\rho\int C_D b(x)\frac{(w - xq)^3}{|w - xq|} + u^2 Z_d\delta_s$$

$$(2.71)$$

$$(-mx_G - M_{\dot{w}})\dot{w} + (I_y - M_{\dot{q}})\dot{q} = (M_q - mx_G)uq + M_wuw - (x_GW - x_BB) - (z_GW - z_BB)\theta$$

$$-\frac{1}{2}\rho\int C_D b(x)\frac{(w - xq)^3}{|w - xq|}xdx + u^2 M_d\delta_s + (L_2\delta w_2 - L_1\delta w_1) \qquad (2.72)$$

$$\dot{z} = -u\theta + w \qquad (2.73)$$

and in matrix form,

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & (m - Z_{\dot{w}}) & -(mx_G + Z_{\dot{q}}) & 0 \\ 0 & (-mx_G - M_{\dot{w}}) & (I_y - M_{\dot{q}}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (2.74)$$

$$A_o = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & Z_wu & (m + Z_q)u & 0 \\ -(z_GW - z_BB) & M_wu & (M_q - mx_G)u & 0 \\ -u & 1 & 0 & 0 \end{bmatrix} \qquad B_o = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & Z_du^2 \\ -L_1 & L_2 & M_du^2 \\ 0 & 0 & 0 \end{bmatrix} \qquad (2.75)$$

16

$$U = \begin{bmatrix} \delta w_1 \\ \delta w_2 \\ \delta_s \end{bmatrix} \qquad state\_variables = \begin{bmatrix} \theta \\ w \\ q \\ z \end{bmatrix} \qquad (2.76)$$

where ;

$$\dot{x} = M^{-1}A_o x + M^{-1}B_o U \qquad (2.77)$$

III. CONTROL SYSTEM DESIGN

Flight control and weight control compose two main subsystem of a soft grounding system. In flight control the vehicle is kept neutrally buoyant and the plane angles are the control inputs. However in weight control, the flow rates for both balast tanks are controlled with zero forward velocity and plane angle. These two components of the designed control system were explained in following sections.

A. FLIGHT CONTROL

The dynamics of underwater vehicles are described by highly nonlinear systems of equations with uncertain coefficients and disturbances that are difficult to measure. An automatic controller for this kind of vehicle must satisty two conflicting requirements: First, it must be sophisticated enough to perform its mission in an open ocean enviroment with ever-changing vehicle/environment interactions. Second, it must be simple enough to achieve real-time control without nonessential computational delays. Sliding mode control theory yields a design that fulfills the above requirements. It provides accurate control of nonlinear systems despite unmodeled system dynamics and disturbances. Furthermore, a sliding mode controller is easy to design and implement. A very effective sliding mode controller can be developed from the linearized equations of motion for an underwater vehicle [Ref. 7].

The sliding mode control design problem can be stated as follows:

Given the system;

$$\dot{x} = f(x) + g(x)u \tag{3.1}$$

where the state vector equation is,

$$x = \begin{bmatrix} \theta \\ w \\ q \\ z \end{bmatrix} \tag{3.2}$$

19

Choose the Lypunov functions :

$$V(x) = \frac{1}{2}[\sigma(x)]^2 \ , \tag{3.3}$$

where

$$\sigma(x) = s^T x \ . \tag{3.4}$$

The scalar function $\sigma(x)$ can be viewed as a weighted sum of the errors in the states x. For stability, it is desired the time derivative of V(x) to be negative,

$$\dot{V}(x) = \sigma\dot{\sigma} \ < 0 , \tag{3.5}$$

This can be achieved if

$$\sigma\dot{\sigma} = -\eta^2|\sigma| \ , \tag{3.6}$$

which means that

$$\dot{\sigma} = -\eta^2 \text{sign}(\sigma) \tag{3.7}$$

Using $\sigma(x) = s^T x$, we get

$$\dot{\sigma} = s^T\dot{x} = s^Tf(x) + s^Tg(x)u = -\eta^2\text{sign}(\sigma) \ , \tag{3.8}$$

and solving for u, the control law is obtained,

$$u = -[s^Tg(x)]^{-1}s^Tf(x) - [s^Tg(x)]^{-1}\eta^2\text{sign}(\sigma) \ , \tag{3.9}$$

In this control law, the first term is nonlinear state feedback, and the second term is a switching control law. The term η^2 is an arbitrary positive quantity, we usually select it such that $\dot{V}$ is negative even in the presence of modeling errors and disturbances. The above control law guarantees stability of $\sigma(x) = 0$, or $s^Tx = 0$. It is necessary to find s. If $\sigma(x) = 0$, the system becomes

$$u = -[s^Tg(x)]^{-1}s^Tf(x) \ , \tag{3.10}$$

and

$$\dot{x} = f(x) - g(x)\left[s^T g(x)\right]^{-1} s^T f(x) \ . \tag{3.11}$$

when the system is linearized,

$$\dot{x} = Ax + Bu \tag{3.12}$$

The previously developed linear state matrix equation is,

$$
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & (m - Z_{\dot{w}}) & -(mx_G + Z_{\dot{q}}) & 0 \\
0 & (-mx_G - M_{\dot{w}}) & (I_y - M_{\dot{q}}) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\dot{\theta} \\
\dot{w} \\
\dot{q} \\
\dot{z}
\end{bmatrix}
=
\begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & Z_w u & (m + Z_q)u & 0 \\
-(z_G W - z_B B) & M_w u & (M_q - mx_G)u & 0 \\
-u & 1 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\theta \\
w \\
q \\
z
\end{bmatrix}
$$

$$
+
\begin{bmatrix}
0 \\
Z_d u^2 \\
M_d u^2 \\
0
\end{bmatrix} \delta \tag{3.13}
$$

$$
M =
\begin{bmatrix}
1 & 0 & 0 & 0 \\
0 & (m - Z_{\dot{w}}) & -(mx_G + Z_{\dot{q}}) & 0 \\
0 & (-mx_G - M_{\dot{w}}) & (I_y - M_{\dot{q}}) & 0 \\
0 & 0 & 0 & 1
\end{bmatrix} \tag{3.14}
$$

$$
A_o =
\begin{bmatrix}
0 & 0 & 1 & 0 \\
0 & Z_w u & (m + Z_q)u & 0 \\
-(z_G W - z_B B) & M_w u & (M_q - mx_G)u & 0 \\
-u & 1 & 0 & 0
\end{bmatrix}
\qquad
B_o =
\begin{bmatrix}
0 \\
Z_d u^2 \\
M_d u^2 \\
0
\end{bmatrix} \tag{3.15}
$$

where $W = W_o$ since the vehicle is neutrally buoyant during the flight. The above can be written

as,

$$\dot{x} = Ax + Bu \tag{3.16}$$

but the dynamics and input matrices should be replaced by

$$A = M^{-1}A_o , \text{ and } B = M^{-1}B_o \tag{3.17}$$

and the system is controlled with sliding mode controller where

$$u = -(s^T B)^{-1} s^T A x - (s^T B)^{-1} \eta \, \text{sat} \, \text{sgn}(s^T x / \phi) \tag{3.18}$$

The closed loop dynamics matrix is

$$A_c = A - B(s^T B)^{-1} s^T A = A - Bk \qquad (3.19)$$

then

$$k = (s^T B)^{-1} s^T A \Rightarrow s^T Bk = s^T A \Rightarrow s^T A - s^T Bk = 0 \quad , \qquad (3.20)$$

so the control law becomes ,

$$u = -kx - (s^T B)^{-1} \eta \, sat \, sgn(s^T x / \phi) \qquad (3.21)$$

The gain vector **k** can be found easily by using Matlab. The Matlab command **place** accepts as inputs the A and B matrices along with a vector of the desired closed loop poles and returns the vector k.

B.   WEIGHT CONTROL

In the second part of the system, the balast tanks were used to ground the vehicle on the ocean floor. The vehicle's grounding can be simulated by adding weight proportionally to both tanks at constant flow rate and by using zero plane angle ($\delta_s = 0$) with those four state variables ($[\theta, w, q, z]'$) defined previously. It is needed to add weight proportionally to eliminate the moment effect since these balast tanks are not located in the same distance from the center of gravity. As it can be seen from the Figure 3-1, $L_1 > L_2$.
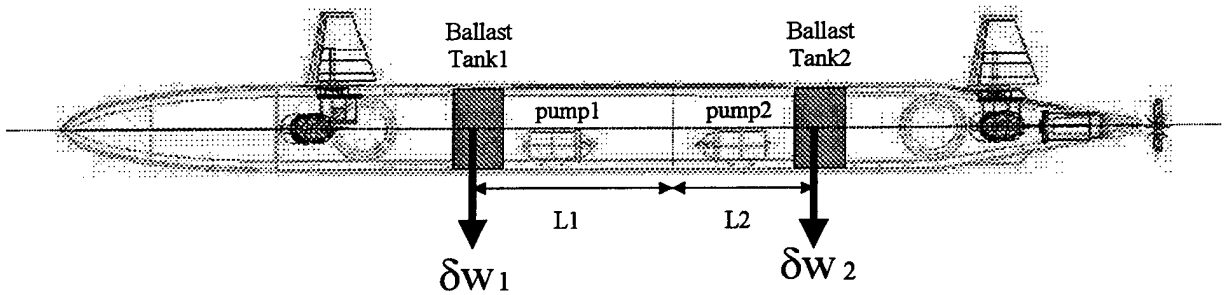

Figure 3-1. The Location of Ballast Tanks

To get zero moment,

$$\delta w_2 = \frac{L_1}{L_2} \delta w_1 \qquad (3.22)$$

If the ballast pumps only pump water into the tanks at fixed rate, there is no control on pitch or depth rate and pitch angles can develop large values. It is proposed therefore, that in order to keep the pitch angle within limits, flow rate of each tank should be controlled separately. This can be achieved by defining two more states to be added to those existing four states. And further these two states are,

$$\begin{aligned} \delta\dot{w}_1 &= f_1 \\ \delta\dot{w}_2 &= f_2 \end{aligned} \qquad (3.23)$$

where $\delta\dot{w}_i$ represents change of weight in tank i and f_i represents flow rate of pump i. So the nonlinear equations of motion become :

$$\dot{\theta} = q \qquad (3.24)$$

$$(m - Z_{\dot{w}})\dot{w} + (-mx_G - Z_{\dot{q}})\dot{q} = (m + Z_q)Uq + mz_G q^2 + Z_w Uw + (W - B)\cos\theta + U^2(Z_{\delta_s} + \alpha Z_{\delta_b})\delta_s$$

$$-\frac{1}{2}\rho\int_{tail}^{nose} C_D b(x)\frac{(w - xq)^3}{|w - xq|}dx \qquad (3.25)$$

$$(-mx_G - M_{\dot{w}})\dot{w} + (I_y - M_{\dot{q}})\dot{q} = (M_q - mx_G)Uq - mz_G wq + M_w Uw - \frac{1}{2}\rho\int_{tail}^{nose} C_D b(x)\frac{(w - xq)^3}{|w - xq|}xdx$$

$$-(x_G W - x_B B)\cos\theta - (z_G W - z_B B)\sin\theta + U^2(M_{\delta_s} + \alpha M_{\delta_b})\delta_s \qquad (3.26)$$

$$\dot{z} = -U\sin\theta + w\cos\theta \qquad (3.27)$$

$$\begin{aligned} \delta\dot{w}_1 &= f_1 \\ \delta\dot{w}_2 &= f_2 \end{aligned} \qquad (3.28)$$

During the grounding operations, plane angles (δ_s) will be zero. Removing related terms, the new equations become,

23

$$\dot{\theta} = q \tag{3.29}$$

$$(m - Z_{\dot{w}})\dot{w} + (-mx_G - Z_{\dot{q}})\dot{q} = (m + Z_q)Uq + mz_G q^2 + Z_w Uw + (\delta w_1 + \delta w_2)\cos\theta$$

$$-\frac{1}{2}\rho \int\limits_{tail}^{nose} C_D b(x)\frac{(w - xq)^3}{|w - xq|}dx \tag{3.30}$$

$$(-mx_G - M_{\dot{w}})\dot{w} + (I_y - M_{\dot{q}})\dot{q} = (M_q - mx_G)Uq - mz_G wq + M_w Uw - \frac{1}{2}\rho \int\limits_{tail}^{nose} C_D b(x)\frac{(w - xq)^3}{|w - xq|}x\,dx$$

$$-(x_G W_o - x_B B)\cos\theta - (z_G W - z_B B)\sin\theta + (-L_1\delta w_1 + L_2\delta w_2)\cos\theta \tag{3.31}$$

$$\dot{z} = -U\sin\theta + w\cos\theta \tag{3.32}$$

$$\delta\dot{w}_1 = f_1 \tag{3.33}$$

$$\delta\dot{w}_2 = f_2 \tag{3.34}$$

where

$$W = W_o + \delta w_1 + \delta w_2 \tag{3.35}$$

When these equations are linearized,

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
0 & (m - Z_{\dot{w}}) & (-mx_G - Z_{\dot{q}}) & 0 & 0 & 0 \\
0 & (-mx_G - M_{\dot{w}}) & (I_y - M_{\dot{q}}) & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\dot{\theta} \\ \dot{\omega} \\ \dot{q} \\ \dot{z} \\ \delta\dot{w}_1 \\ \delta\dot{w}_2
\end{bmatrix}
$$

$$
=
\begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 \\
0 & Z_w u & (m + Z_q)u & 0 & 1 & 1 \\
-(z_G W_o - z_B B) & M_w u & (M_q - mx_G)u & 0 & -L_1 & L_2 \\
-u & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\theta \\ \omega \\ q \\ z \\ \delta w_1 \\ \delta w_2
\end{bmatrix}
+
\begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
1 & 0 \\
0 & 1
\end{bmatrix}
\begin{bmatrix}
f_1 \\ f_2
\end{bmatrix}
\tag{3.36}
$$

24

Flow rates for balast tanks are control inputs. In the next step, the control system will be designed to keep the pitch angle and depth rate within the limits during grounding of the vehicle.

C.   WEIGHT CONTROL WITH LINEAR QUADRATIC REGULATOR

The system was given as,

$$\dot{x} = Ax + Bu \qquad (3.37)$$

where the gain matrix K of the optimal control vector,

$$u(t) = -Kx(t) \qquad (3.38)$$

minimizing the performance index gives

$$J = \int_0^\infty (x^T Q x + u^T R u) dt \qquad (3.39)$$

where Q is a positive-definite Hermitian matrix and R is a positive-definite Hermitian matrix. ($u^T R u$) term accounts for the expenditure of the energy of the control signals. The matrices Q and R determine the relative importance of the error and the expenditure of this energy. Q is the state weighting matrix and R penalizes the control effort. It can be choosen relatively small elements of Q compared to R for the control law which will tolerate errors in x with low control effort u. Chosing larger elements of Q compared to R will result in tight control which means small errors will need considerably more control effort. The advantage of using the quadratic optimal control system is that the system will be stable as long as it is conrollable [Ref. 8].

By using the general equations of optimal control to solve the LQR problem gives the Hamiltonian

$$H(x, p, u) = p^T (Ax + Bu) - \frac{1}{2}(x^T Q x + u^T R u) \;, \qquad (3.40)$$

where p is an unknown vector (co-state vector). The necessary conditions for optimality are,

$$\dot{x} = \frac{\partial H}{\partial p} \Rightarrow \dot{x} = Ax + Bu \qquad (3.41)$$

25

$$\dot{p} = -\frac{\partial H}{\partial x} \Rightarrow \dot{p} = -A^T p + Qx \qquad (3.42)$$

$$\frac{\partial H}{\partial u} = 0 \Rightarrow B^T p - Ru = 0 \Rightarrow u = R^{-1} B^T p \qquad (3.43)$$

and the boundry conditions are,

$$[p^T(t_f) + x_f^T F] \delta x_f = 0 \qquad (3.44)$$

where x_f is free, δx_f is arbitrary and F is symmetric and positive-definite weighting matrix. So, the quantity inside the square brackets must be equal to zero and this produces a new form of boundary condition,

$$p(t_f) = -Fx(t_f) \qquad (3.45)$$

Now, the equations that have to be solved can be listed as follows,

$$
\begin{aligned}
\dot{x} &= Ax + BR^{-1}B^T p, \\
\dot{p} &= Qx - A^T p, \\
x(t_o) &= x_o, \\
p(t_f) &= Fx(t_f).
\end{aligned}
\qquad (3.46)
$$

From above equations, p(t) can be calculated and this will provide u as a function of time from $u = R^{-1}B^T p(t)$. By using Kalman's idea, p(t) can be defined as,

$$p(t) = -S(t)x(t) \qquad (3.47)$$

where S(t) is a symmetric positive-definite matrix. Also,

$$\dot{p} = -\dot{S}x - S\dot{x} = -\dot{S}x - S(Ax + BR^{-1}B^T p) \qquad (3.48)$$

and ,

$$Qx - A^T p = -\dot{S}x - SAx - SBR^{-1}B^T p \qquad (3.49)$$

$$-\dot{S}x = (A^T S + SA - SBR^{-1}B^T S + Q)x \qquad (3.50)$$

and for this to be true for all x,

26

$$-\dot{S} = A^TS + SA - SBR^{-1}B^TS + Q \quad \text{(with } S(t_f) = F\text{)} \tag{3.51}$$

This is called Riccati matrix differential equation and *S(t)* can be obtained by backwards integration of this equation. Since $u = R^{-1}B^Tp(t)$ the closed loop optimal control law can be found by

$$u = R^{-1}B^TSx \tag{3.52}$$

where *S* can be found by solving the algebraic Riccati equation for the positive-definite *S*,

$$A^TS + SA - SBR^{-1}B^TS + Q = 0 \tag{3.53}$$

In Matlab, the command

$$lqr(A,B,Q,R)$$

solves the continuous-time, linear, quadratic regulator problem and the associated Riccati equation. This command calculates the optimal feedback gain matrix K for control law which minimizes the performance index.

D.   GROUNDING WITH VERTICAL THRUSTERS

The bladed thrusters are the essential elements of improved vehicle positioning systems. With automatic position control, the thrusters enable important scientific and industrial tasks such as automatic docking, station keeping, precise surveying, inspection, sample gathering and manipulation. Incorporating precise models of thruster dynamics into the feedback control systems of marine vehicles promises improved vehicle positioning [Ref. 9].

Most small-to-medium sized underwater vehicles are powered by electric motors driving propellers mounted in ducts. The propeller is mounted in a duct or shroud in order to increase the static and dynamic efficiency of the thruster. Thrusters are subject to serious degradation due to axial and cross flow effects. Axial flow effects can be reasonably approximated by the modeling of the thruster unit alone, the velocity of the fluid entering the thruster shroud effectively changes the angle of attack of the propeller , thus altering the force produced. Cross flow effects are much

more difficult to model and are highly dependent on the position of the thruster on the vehicle. The amount of force produced by the thruster will reduce the overall gain of a control system unless these effects are specifically in the controller design[Ref. 10].

For this work, NPS Phoenix vehicle is taken as an example. Figure 3-2 shows the locations of vertical and horizontal thrusters on the vehicle.Those four tubes represents the thruster shrouds. In Figure 3-3, the vertical thruster tubes can be seen throughout the vehicle. Thruster blades are located close to the bottom of those tubes.

Thruster moment and force equations were developed by Louis L.Whitcomb and Dana R.Yoerger [Ref. 9]. On that paper, the control system for these thrusters was also discussed. But in this study, these thruster force and moments were assumed as some constant parameters and also no control law was developed to control them. Since the main element for grounding is the weight control, the thrusters were just used as auxillary elements of this procedure in order to increase depth rate. By using thrusters in addition to the weight control, following changes should be done to heave and pitch rate equations,

$$(m - Z_{\dot{w}})\dot{w} + (-mx_G - Z_{\dot{q}})\dot{q} = (m + Z_q)Uq + mz_G q^2 + Z_w Uw + (\delta w_1 + \delta w_2)\cos\theta$$

$$-\frac{1}{2}\rho \int_{tail}^{nose} C_D b(x)\frac{(w - xq)^3}{|w - xq|}dx + Z_{thruster} \tag{3.54}$$

$$(-mx_G - M_{\dot{w}})\dot{w} + (I_y - M_{\dot{q}})\dot{q} = (M_q - mx_G)Uq - mz_G wq + M_w Uw - \frac{1}{2}\rho \int_{tail}^{nose} C_D b(x)\frac{(w - xq)^3}{|w - xq|}xdx$$

$$-(x_G W_o - x_B B)\cos\theta - (z_G W - z_B B)\sin\theta + (-L_1 \delta w_1 + L_2 \delta w_2)\cos\theta - M_{thruster}q \tag{3.55}$$

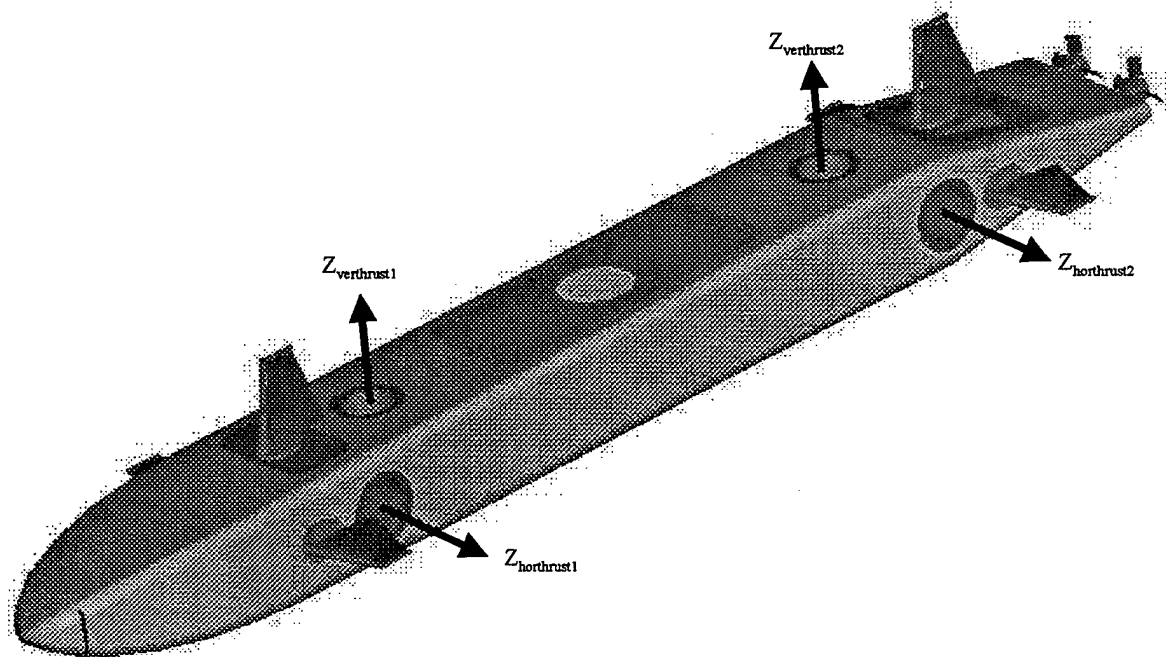where $Z_{thruster}$ and $M_{thruster}$ are thruster force and thruster moment respectively.

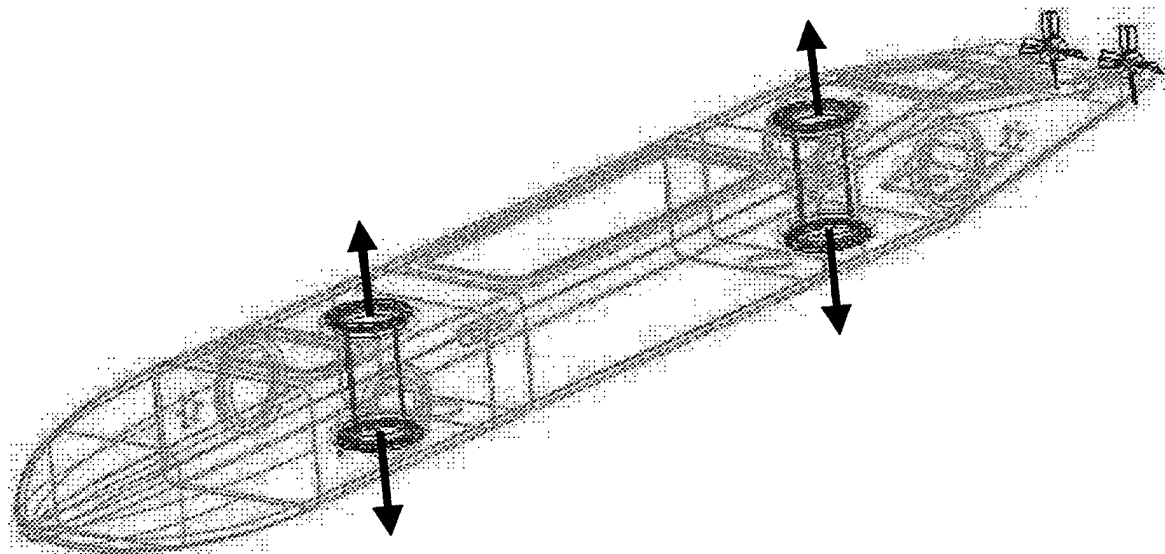**Figure 3-2. Horizontal and Vertical Thrusters of NPS Phoenix AUV**

Figure 3-3. Vertical Thrusters and Thruster Shrouds of NPS Phoenix AUV

29

IV.   SIMULATION

For the realistic simulation of the grounding system, some other control functions were used in addition to the flight and weight controls. The system was simulated by using Matlab's **Simulink** program. For the dynamic control, Matlab's s-**function** was used. This tool gives us the opportunity to use the memory dynamically. In fact, it is nothing different than a ODE solver. In that block, the state equations were solved by using Matlab's **ode45** function with variable step size.

As depicted in Figure 4-1, two main components of control system are flight control and weight control, and they also form two main loops of simulation.
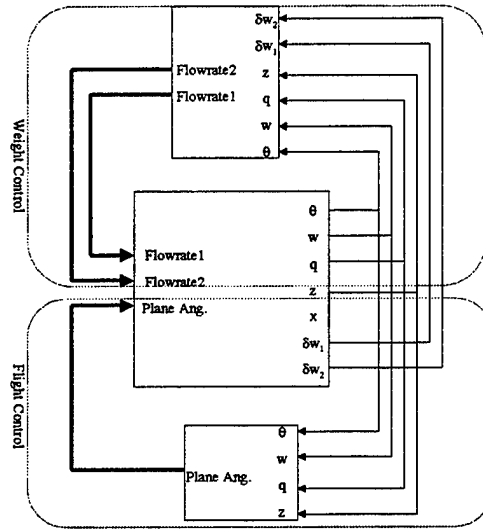
Figure 4-1. Dynamic Controller with Weight and Flight Controllers

For the simulation of longitudinal motion, in addition to the states used in the control laws, X was also included. But this term was not used in control design. X was defined as,

$$X = u.\cos\theta + w\sin\theta + U_{cz} \qquad (4.1)$$

where u is the forward speed and U_{cz} is current.

The design of weight and flight controls were explained in previous chapter. Z_{com} is the commanded flight depth. The sliding mode controller calculates the control surfaces from the

error which is the difference between the commanded and actual depths. The switch from the flight control to the weight control is decided by the forward speed command produced by speed control unit. This unit has two parts: Speed control and secondary control units.

The speed control block seen in Figure 4-2 reduces the forward velocity to 0.1 ft/sec when the vehicle reaches to the commanded flight depth. But this is not a sudden decrease in the speed. Since the longitudinal position was also commanded by X_{com} block, the control unit calculates the necessary break distance in order to reach that location. The break distance was determined from a known deceleration in the longitudinal direction and the time needed to ground from the flight depth. As seen on the Figure 4-2, the input parameters to the speed control block are current and commanded depths, current and commanded longitudinal locations, commanded forward speed and ground depth; the output of the block is the controlled forward speed.

Speed Control

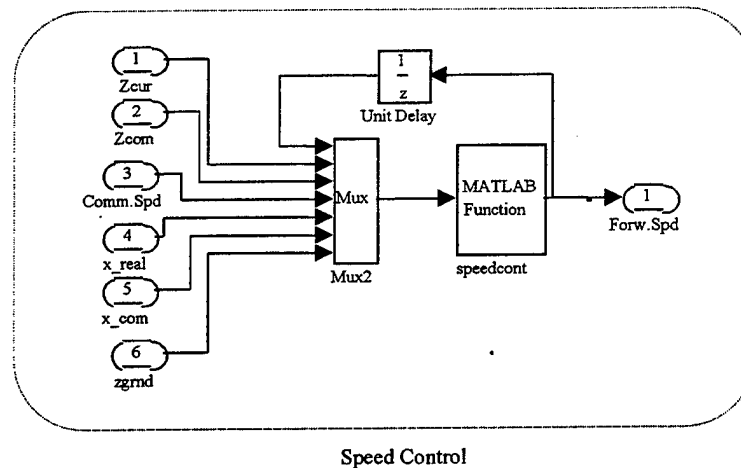**Figure 4-2. Speed Control Block**

As mentioned above, the main control input for the speed control block is the depth. When the depth error (Z_{com} − Z_{curr}) becomes smaller than a certain number, the deceleration procedure starts. One problem with this method is at the beginning of the grounding, the depth error starts to increase again which means that the speed control will increase the forward

32

velocity until it is equal to commanded speed. To prevent this, the secondary control unit seen in

Figure 4-3 was designed:
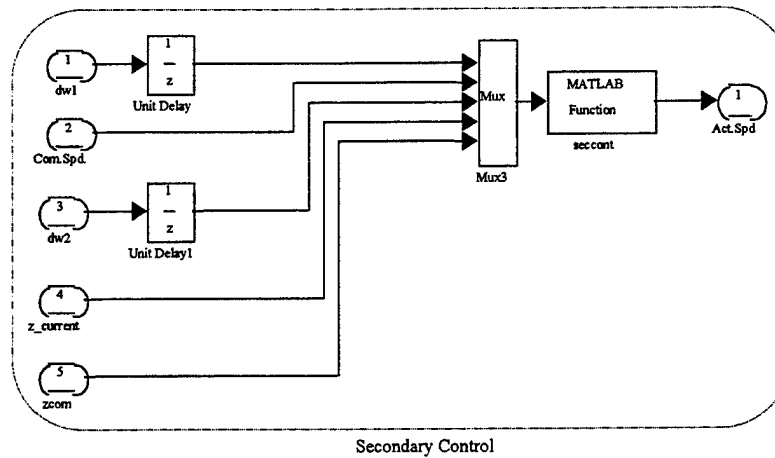
Secondary Control

Figure 4-3 Secondary Speed Control Block

Two of the five inputs of this block are the weights in both ballast tanks. If the weight in one or both of those two tanks are different than zero, then the output of the secondary control becomes 0.1 (forward speed). Otherwise, the block passes the value coming from the speed controller without changing.

The speed command produced by the secondary controller provides the switch from flight to the grounding procedures. Figure 4-4 shows secondary control, plane angle controller and weight controller blocks.

Figure 4-4. The Relationship of Secondary Control Block with Plane Angle and

Weight Controller Blocks

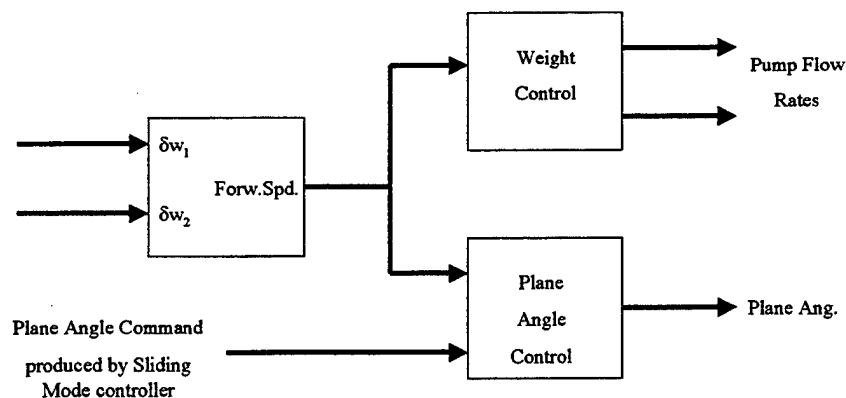When the forward velocity is reduced to 0.1 ft/sec, the weight controller starts to produce flow rate commands by using the methods described in Chapter III. The plane angle control block passes the value coming from the sliding mode controller (flight control block) without changing. But with the reduced speed, it makes the plane angle zero. So this ends the effect of flight control on the vehicle, in other words, the weight controller gets the control on the vehicle's motion.

Figure 4-5 shows all functions mentioned above with flight and weight controllers as a complete Simulink diagram. The flow chart in Figure 4-6 shows the logic of the procedure. The Matlab programs used in the simulation are presented in Appendix A.
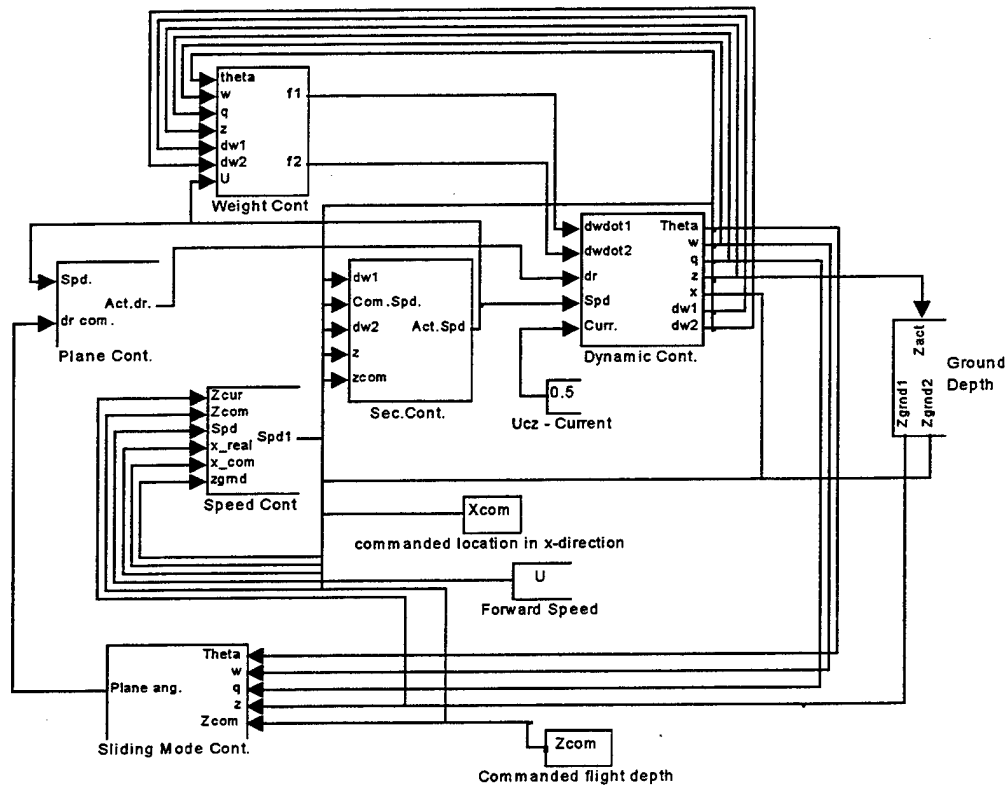
Figure 4-5. Simulink Block Diagram Used for Simulation

Flight Cont.

$U_{cz} = 0.5$ ft/sec
$X_{comm} = 250$ ft
$Z_{comm} = 4$ ft
$\delta w_1, \delta w_2 = 0$

Speed Cont

No ← $Z = Z_{comm}$ → Yes

No ← $X = X_{comm} - X_{brake}$ → Yes

$U = U - 0.1$

Plane Ang.Cont.

No ← $U \leq 0.1$ → Yes

Plane Ang. = 0
Weight Control

No ← $Z = Z_{ground}$ → Yes

Weight Cont.

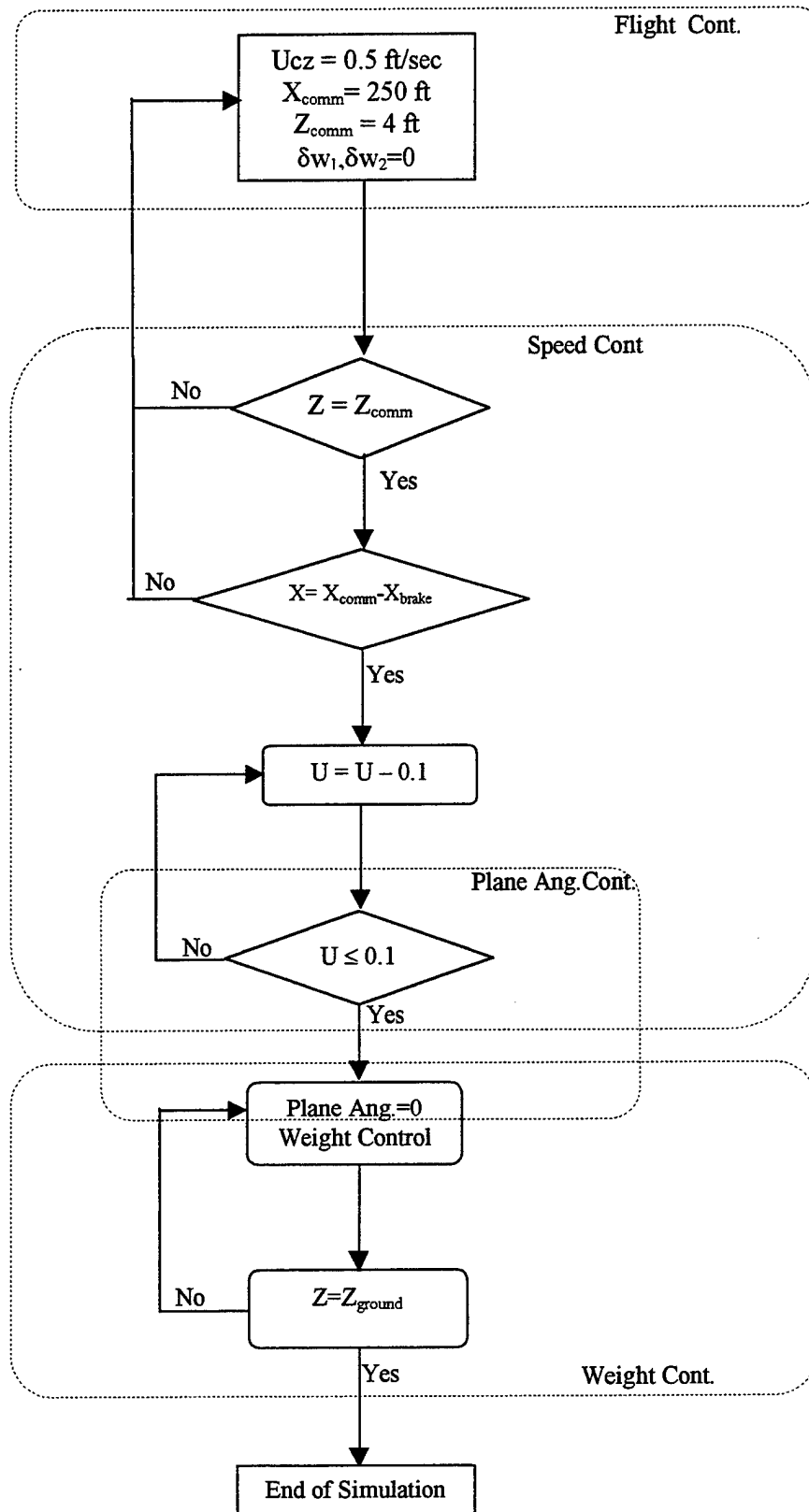End of Simulation

Figure 4-6. Flow Chart for Control Steps

35

V.   RESULTS AND DISCUSSION

A.   INTRODUCTION

In previous chapters, the mathematical models of the control system were developed. To prove the validity of the flight and weight controllers, the system was simulated by using the parameters of NPS Phoenix AUV. These parameters are presented in Appendix B. First the AUV was controlled by flight controller (sliding mode control) during its diving from surface to a commanded flight depth. Second, the flight controller and the different cases of weight controller were simulated on the vehicle.

B.   FLIGHT CONTROL

In the NPS Phoenix vehicle , there are four vertical control planes powered by servo motors. Using the definitions in Chapter II and Chapter III with the parameters of NPS Phoenix AUV and a nominal speed of 4 ft/sec, A and B matrices becomes

$$
A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0.0155 & -3.4119 & -0.9247 & 0 \\ -0.1086 & 0.6027 & -0.9667 & 0 \\ -4 & 1 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ -0.7139 \\ -0.0123 \\ 0 \end{bmatrix} \tag{5.1}
$$

By choosing the poles as p = [-2, -2.1, -2.2, 0], the vector k was calculated,

$$
k = place(A,B, [-2, -2.1, -2.2, 0]) \tag{5.2}
$$

$$
k = [-18.3998, -2.4217, -15.6369, 0] \tag{5.3}
$$

A_C is calculated from

$$
A_C = A - Bk \tag{5.4}
$$

$$
A_c = \begin{bmatrix} 0 & 0 & 1 & 0 \\ -13.1205 & -5.1408 & -12.0882 & 0 \\ -0.3352 & 0.5729 & -1.1592 & 0 \\ -4.0 & 1 & 0 & 0 \end{bmatrix} \tag{5.5}
$$

37

The eigenvector of $A^T{}_C$ for the pole at the origin is the sliding surface

$$s = [-0.8733 \ ; \ -0.0272 \ ; \ -0.4692 \ ; \ 0.1287] \qquad (5.6)$$

as a result with $\phi = 0.1$, the control law becomes

$$\delta = -18.3998\theta - 2.4217\omega - 15.6319q$$
$$- (0.4)\text{sat sgn}\left\{\left[-0.8733\theta - 0.0272\omega - 0.4692q + 0.1287(z - z_{com})\right]/0.1\right\} \qquad (5.7)$$

The motion of the vehicle is restricted to the vertical plane. The motion profiles for depth and pitch have been specified using sliding mode control. For the maneuver, the commanded depth was 4 ft and the vehicle was originally at the surface. As can be seen from the Figure 5-1, during the flight, maximum pitch angle becomes 0.042 rad (~ 2.5 degrees). When the vehicle reaches to the commanded depth as seen in Figure 5-2, the pitch angle becomes zero as expected . The controller produces the dive plane angle command according to the depth error. In the beginning the depth error is large, so the system produces higher values of plane angle command in order to eliminate this error. With the full state feedback, the vehicle responded very well to these commands.

C. ADDING WEIGHT TO BOTH TANKS WITHOUT CONTROL

After completion of the flight to the commanded depth, the vehicle gets water to both tanks in order to become heavy. Figure 5-3 shows the weight increase in the tanks. During grounding, the planes kept at zero degrees as depicted in Figure 5-4. To keep depth rate within limits, the maximum weight pumped in was limited at 5 lb for each tank. With this additional weight, the vehicle sat on the ground (10 ft.) with 0.6 ft/sec depth rate. Eventhough the weight was added proportionally ($\delta w_2 = (L_1 / L_2)\delta w_1$) to get zero momentum effect, there is still some momentum because of the vehicle itself. This momentum was created by the pitch and heave motion of the vehicle. As a result, the pitch angle increases since there is no control on either depth rate or pitch angle. As seen on Figure 5-5 and Figure 5-6, at the end of a 6 ft. drop, the pitch

38

angle becomes 0.6 rad. (35 degrees). This method can be used for very short grounding depths (2-3 ft), but for other cases, it is not recommended since the system is completely unstable.
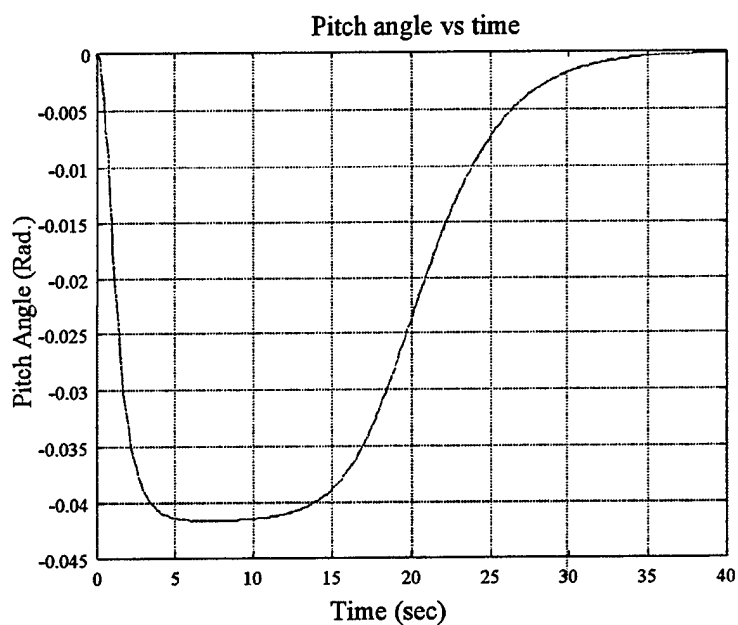

Figure 5-1. Pitch Angle Plot During The Flight

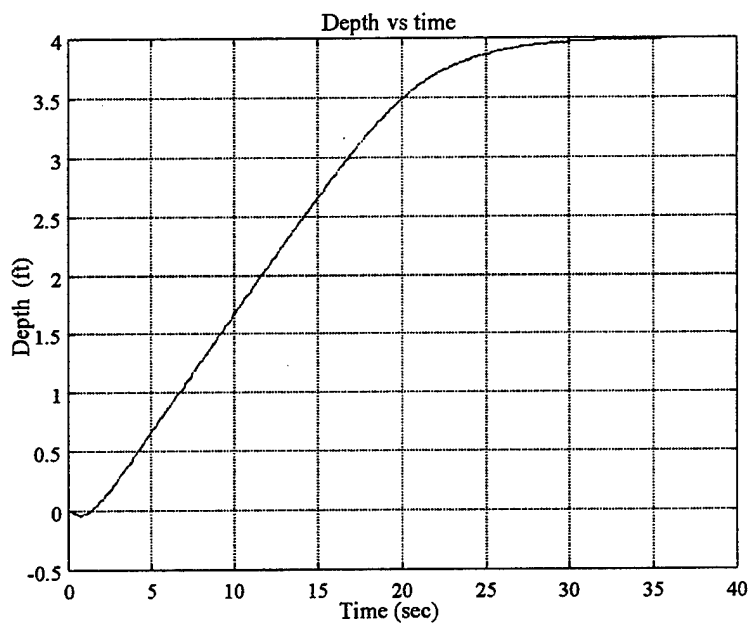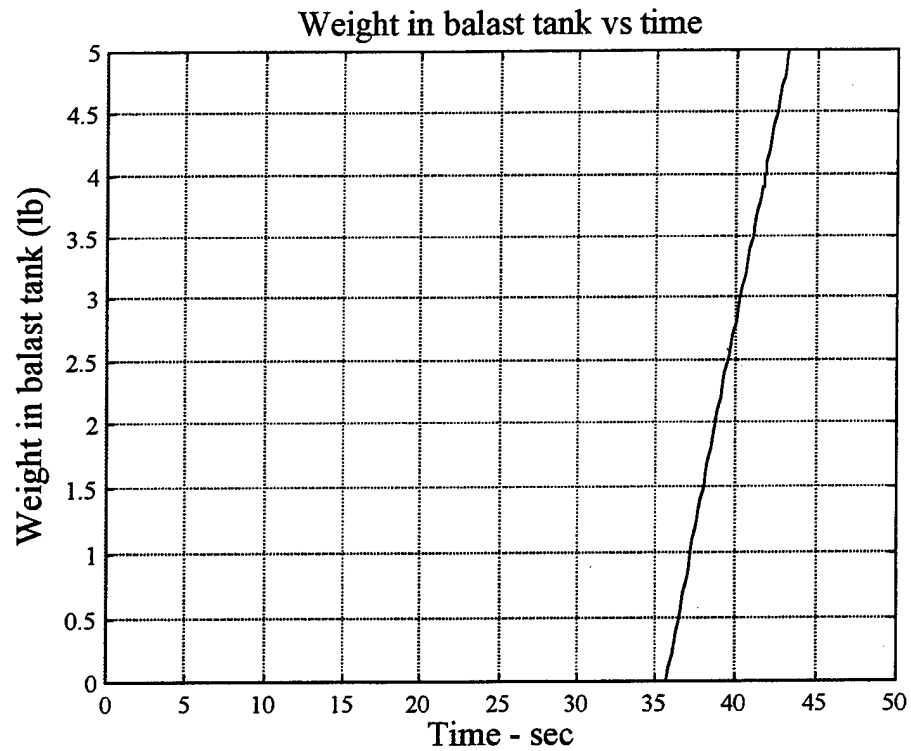**Figure 5-2. Depth Change As A Function Of Time During The Flight.**

Figure 5-3. Weight Addition During Grounding With No Control On Flow Rates

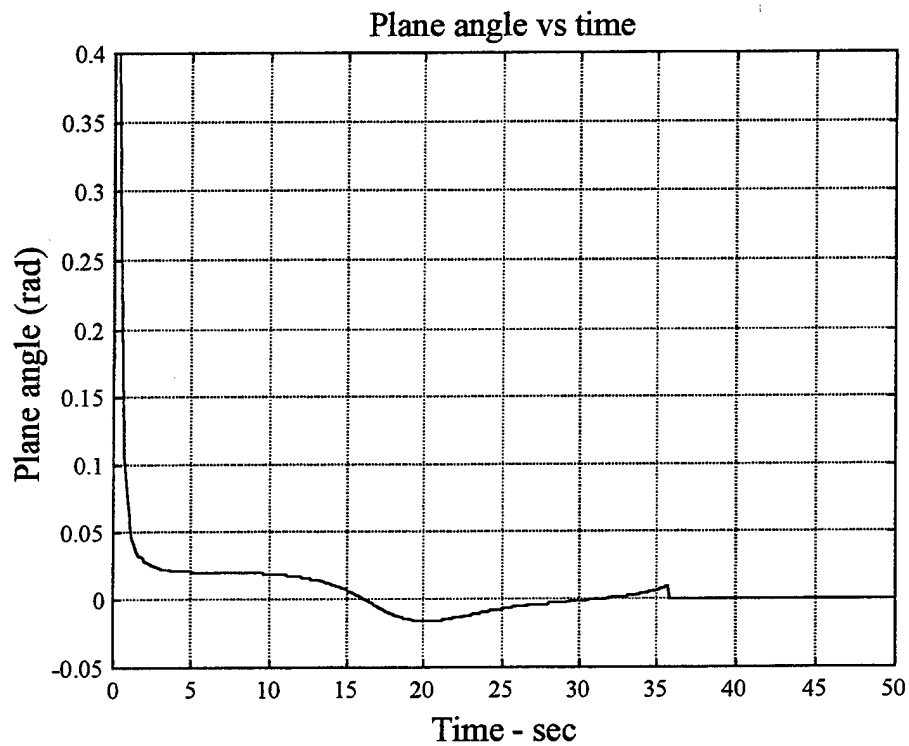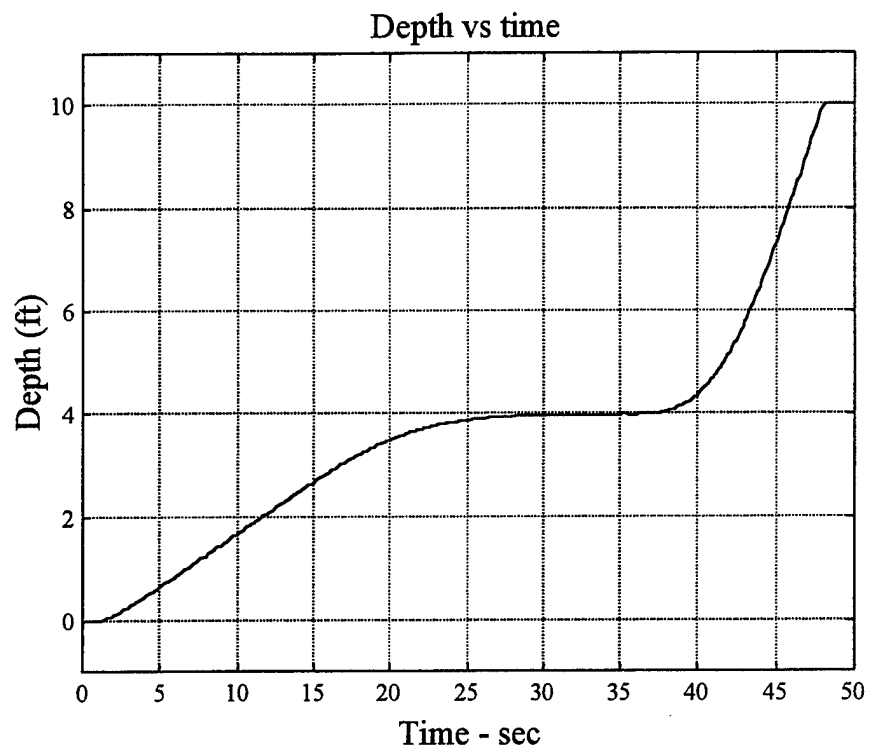**Figure 5-4. Plane Angle During Flight And Grounding**

40

Depth vs time

Figure 5-5. Depth Change During The Flight And Grounding
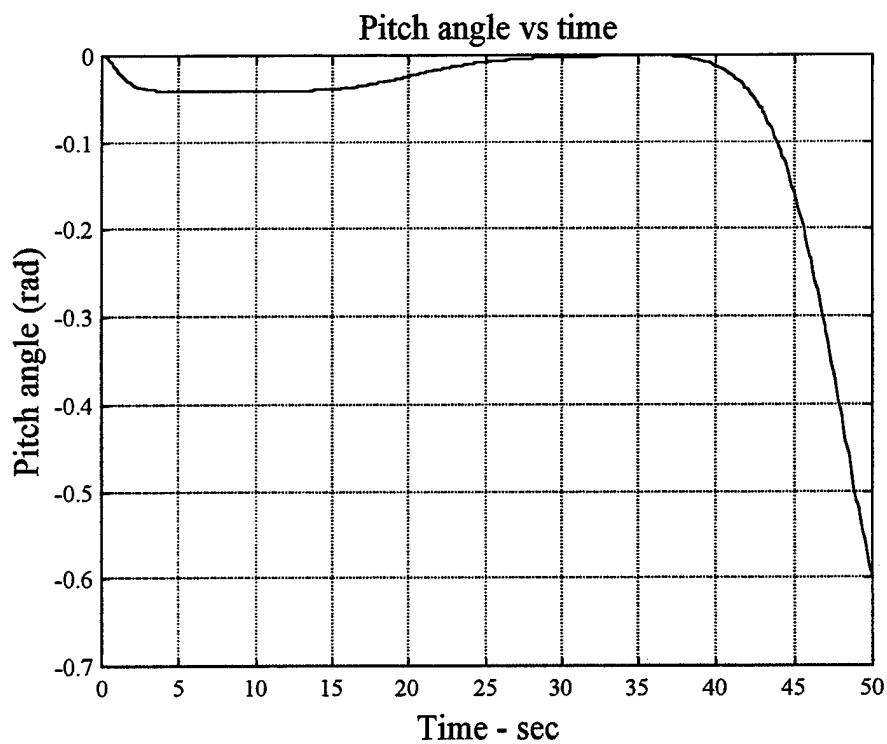
Pitch angle vs time

Figure 5-6. Pitch Angle Change For Flight And Grounding

41

D. WEIGHT CONTROL WITH LINEAR QUADRATIC REGULATOR

In order to keep the depth rate and pitch angle within limits, the linear quadratic regulator technique was used in designing a weight control. The mathematical model for this control system was previously explained in Chapter III. The parameters of NPS Phoenix vehicle was used for simulation. With these known parameters and 0.1 ft/sec forward velocity, A and B matrices become,

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0.0155 & -0.0853 & -0.0231 & 0 & 0.0223 & 0.0196 \\ -0.1086 & 0.0151 & -0.0242 & 0 & -0.0130 & 0.0065 \\ -0.1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad (5.8)$$

In practice, high values of pitch angle (> 15 degrees) are not desired for a safe and stable grounding of the vehicle. So the designed control law should not tolerate too much oscillations of pitch angle. This kind of control law can be provided by choosing larger elements of Q for pitch angle compared with the others. So, Q and R matrices were chosen as follows:

$$Q = \begin{bmatrix} 10^4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 10^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad (5.9)$$

$$R = \begin{bmatrix} 3 \times 10^5 & 0 \\ 0 & 3 \times 10^5 \end{bmatrix} \qquad (5.10)$$

By using **lqr** command in Matlab, control gain matrix K can be obtained,

$$[K,s,m] = lqr(A,B,Q,R) \qquad (5.11)$$

and with the numbers provided,

42

$$K = \begin{bmatrix} 0.0608 & 0.1263 & -0.0526 & 0.0138 & 0.0754 & 0.0365 \\ -0.0322 & 0.1294 & 0.0021 & 0.0120 & 0.0365 & 0.0613 \end{bmatrix} \quad (5.12)$$

Since grounding to the ocean floor from a certain depth desired, the command matrix should be

$x_{com} = [0\ 0\ 0\ z_{gr}\ 0\ 0]'$ where z_{gr} = (ground depth) - (the depth where grounding is started). So the control law becomes ,

$$U = -Kx_{error} \quad (5.13)$$

where

$$x_{error} = x - x_{com} \quad (5.14)$$

The simulation of the system with this control law can be seen in Figure 5-7 through Figure 5-14.

Positive flow rate represents water inlet to the balast tanks, and negative flow represents the opposite. The pumps are not allowed to pump out when there is no water in balast tanks. Figure 5-7 and Figure 5-8 show when the weight in a balast tank and the pump flow rate become zero. This is provided by a simple controller which compares the weight in ballast tank (δw) and flow rate (f). If δw is equal to zero and flow rate is a negative number, than the control input (f) of that pump becomes zero.

At the commanded depth, the speed control unit slows down the vehicle to an almost zero forward velocity (u = 0.1 ft/sec). The speed control unit's other duty is to control the longitudinal position. During the flight, the speed control unit compares the vehicle's location (x) with commanded location (x_{com}) which is a longitutional distance from the original position. When the vehicle is at the commanded depth of flight, a deceleration procedure starts. A simple algorithm was used to calculate the minimum distance needed for deccaleration to reach the commanded location at the end of the grounding. The change of forward velocity due to the depth change can be seen in the Figure 5-13.

The depth rate as seen in Figure 5-11 is very low in this method because of the command given to the weight control. The weight control produces its control values due to the errors which are the differences between the commanded and the actual states. In the above case, only the depth (z) command has a value, the commands for other states are zero. At the end of the simuation, the pitch angle becomes almost zero as seen in Figure 5-12. In the first half of the grounding, Figure 5-9 and Figure 5-10 shows an increase in weight for both tanks, but in the second half, the system tries to make the vehicle neutrally buoyant as expected.

E.   LQR WITH POSITIVE WEIGHT COMMAND

So when the vehicle reaches the ground, there will be almost no water in balast tanks. But for the stability of the grounding, the vehicle should be heavier. For this reason, in addition to the depth command, the weight can also be commanded to increase the depth rate or the weight of the vehicle at the end of the grounding. So, the command vector, x_{com} is changed to ,

$$x_{com} = [0 , 0 , 0 , z_{ground} , \delta w_{com1} , \delta w_{com2}]' \qquad (5.15)$$

where δw_{com} are some positive numbers and represent the command for additional weight and the system was simulated with these new parameters. Figure 5-15 through Figure 5-20 show the plot of this simulation. In this case, pitch angle reaches a maximum value of 0.18 rad (10.31 degrees) which is 4 times greater than the previous simulation.With increasing pitch angle, there is also an increase in the depth rate. The depth rate becomes 0.35 ft/sec which is again almost 4 times greater than the previous simulation. Since the commanded depth for the control law is the depth of the ground, the system tries to make the vehicle neutrally stable to keep the vehicle on that depth by pumping water out of balast tanks. After grounding, pumps should pump water in balast tanks until they are full. Because this additional weight is needed to keep the vehicle sitting on the ground against the current.

Another method for increasing the depth rate is to command with a depth value which is greater than the actual ground depth. Because in the beginning the error will be higher, the weight controller will produce higher values of flow rate.
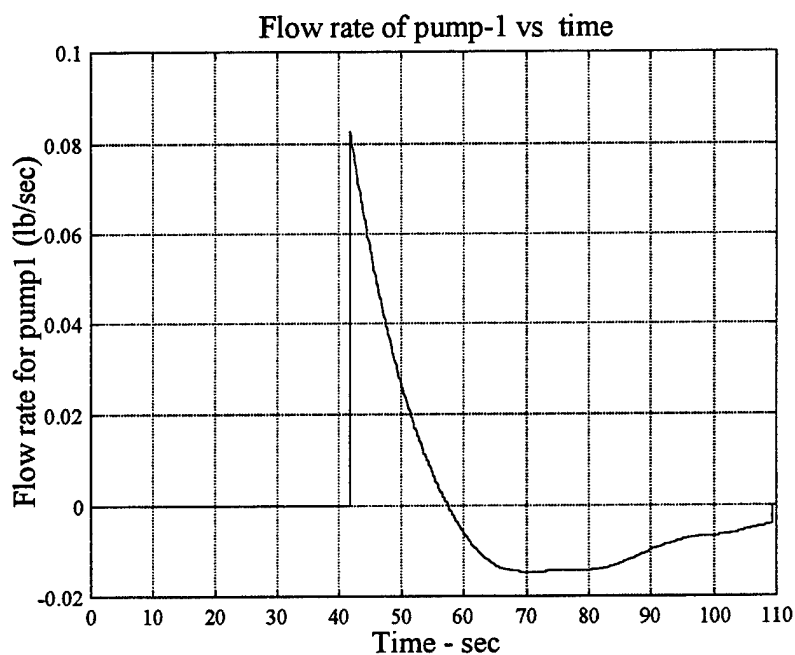
Figure 5-7. Flow Rate As A Function Of Time For Pump-1 With Depth Command Only
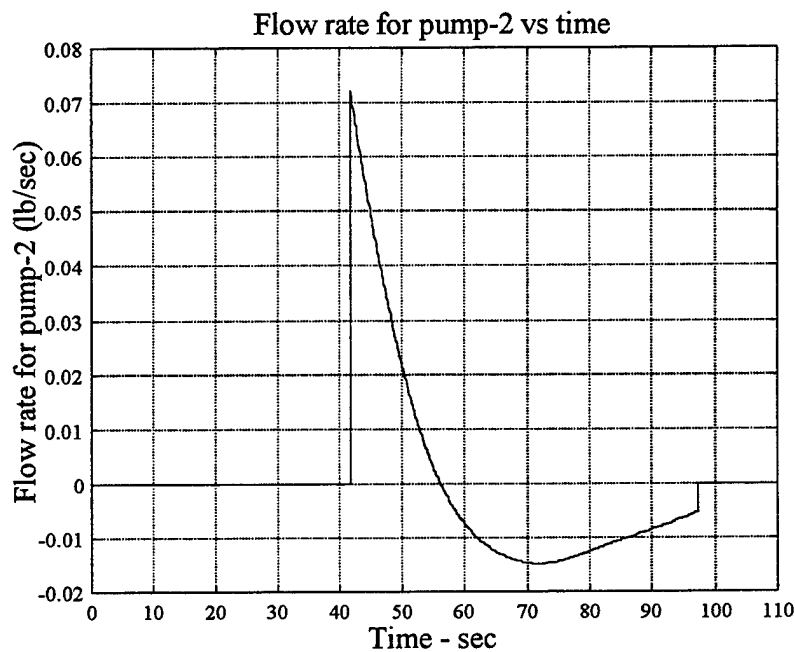
Figure 5-8. Flow Rate As A Function Of Time For Pump-2 With Depth Command Only

45
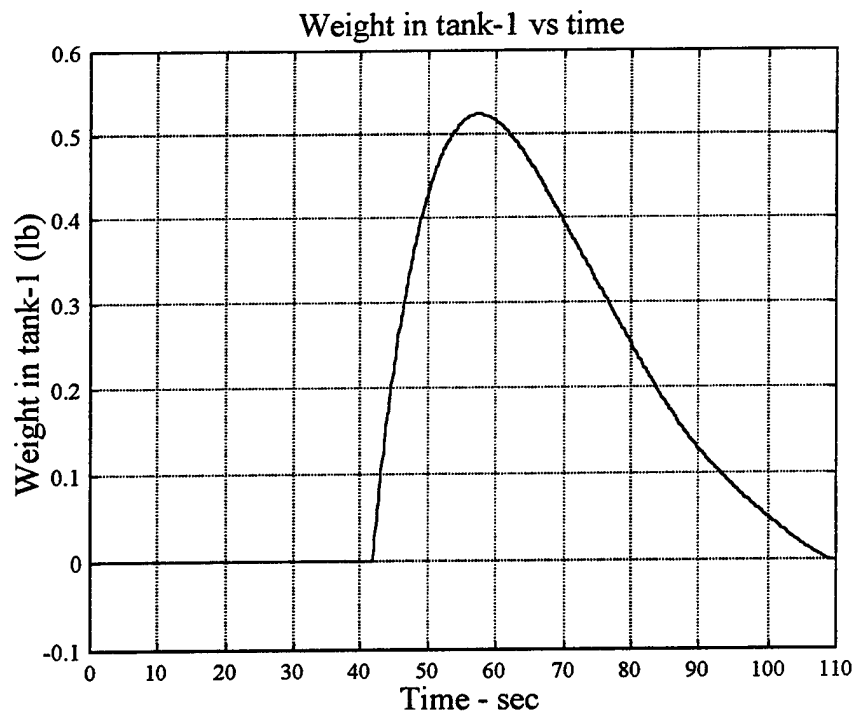
Figure 5-9. Weight Change In Ballast Tank-1 With Depth Command Only
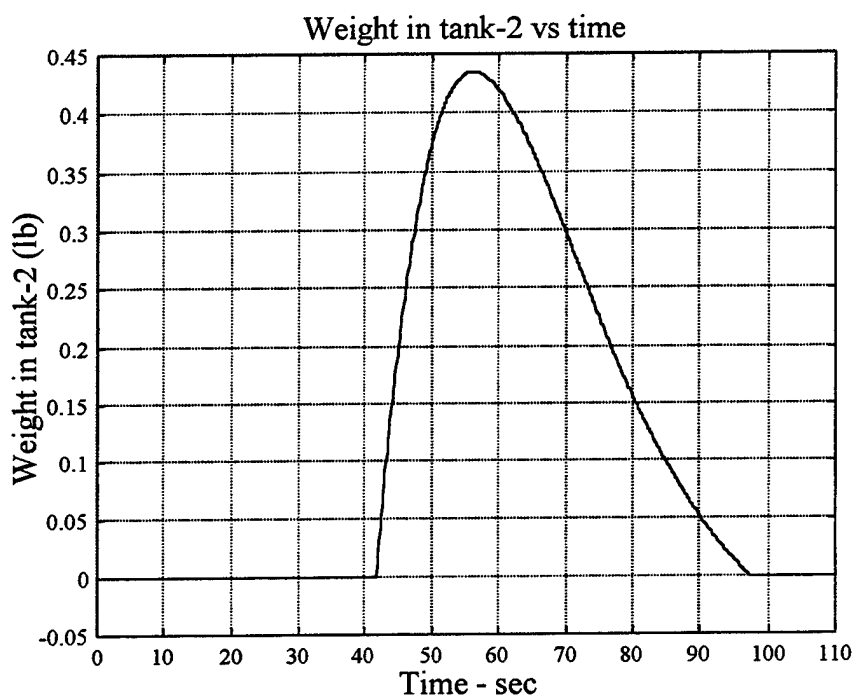
Figure 5-10. Weight Change In Ballast Tank-2 With Depth Command Only

46

Depth vs time
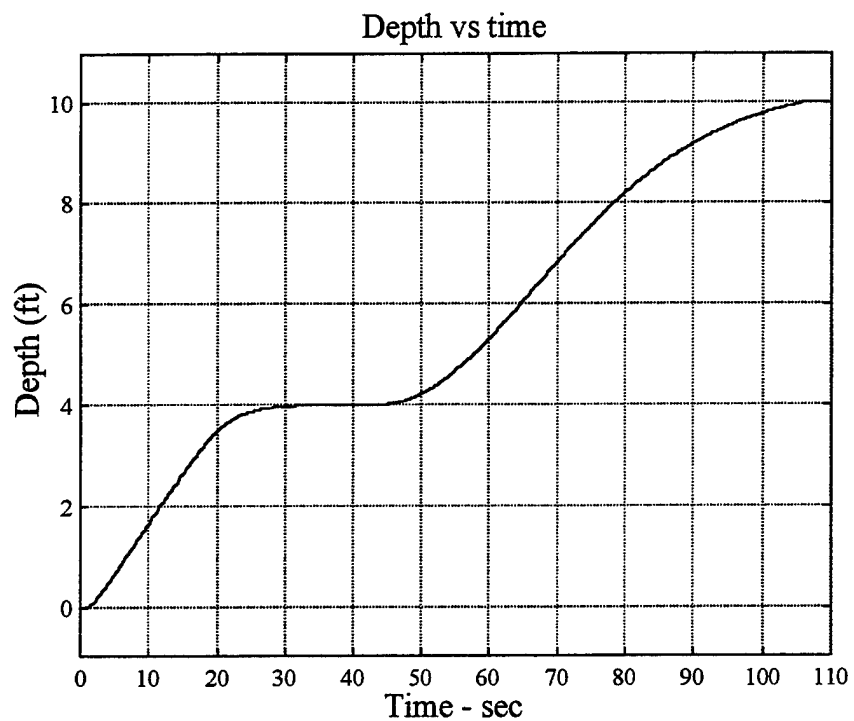
Figure 5-11. Depth Change During Flight And Grounding

Pitch angle vs time
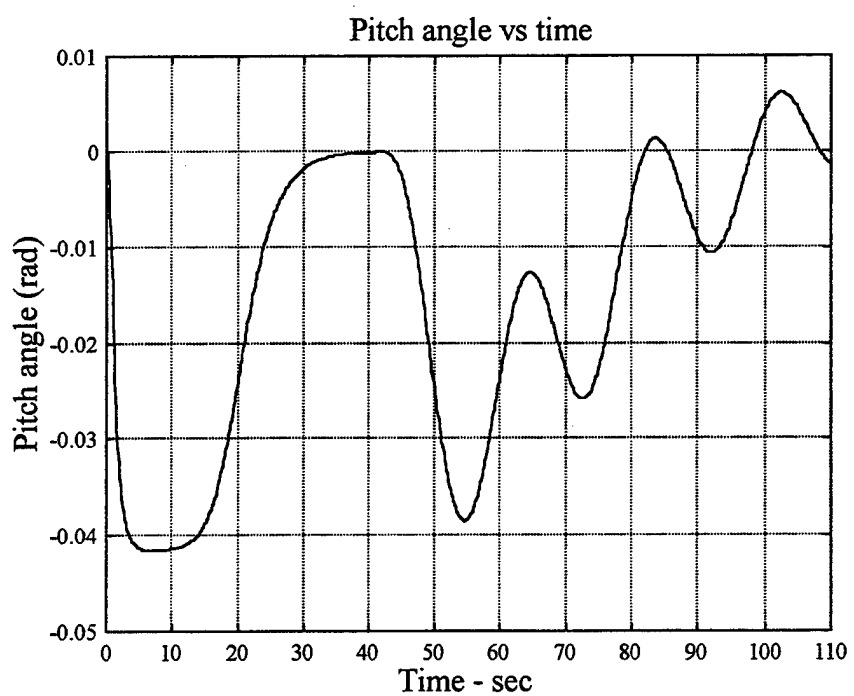
Figure 5-12. Pitch Angle Change With Depth Command Only

Change in depth and forward velocity
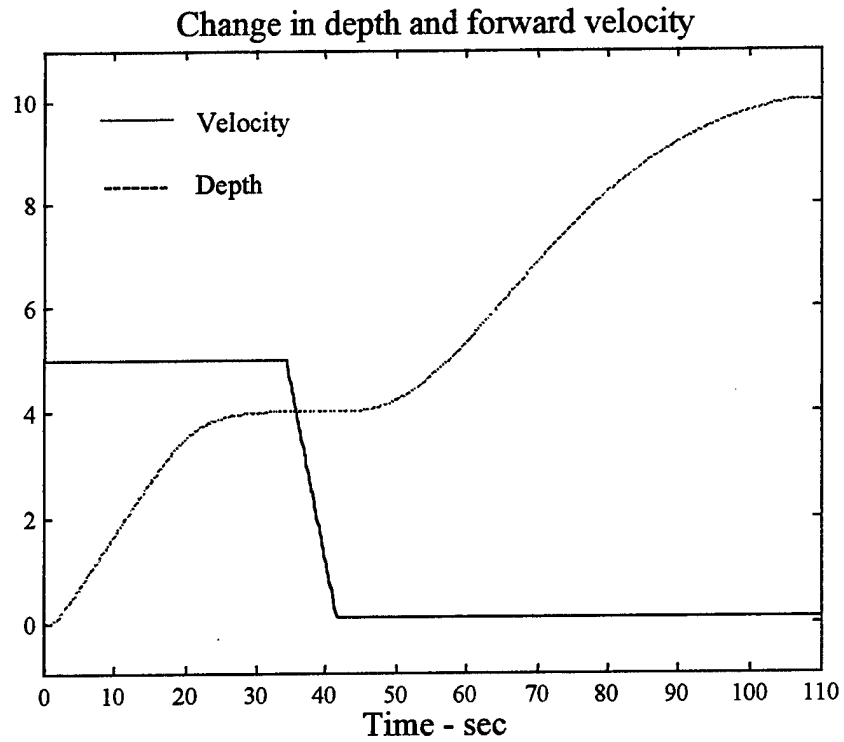
Figure 5-13. Comparison Of Depth And Forward Velocity Change

Distance from the original position vs depth
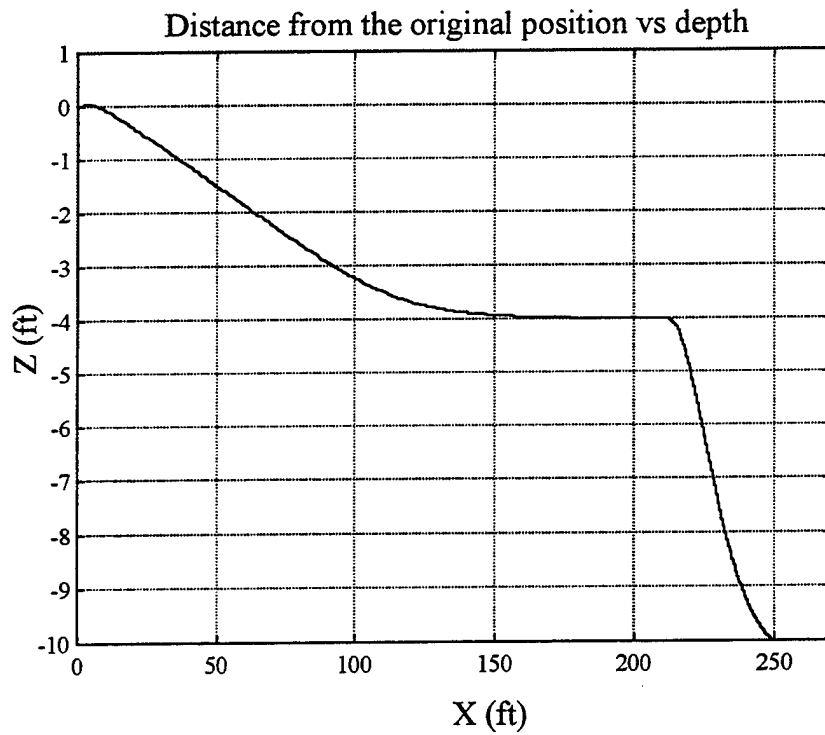
Figure 5-14. Response Of The Vehicle To The Longitudinal Position Command

48
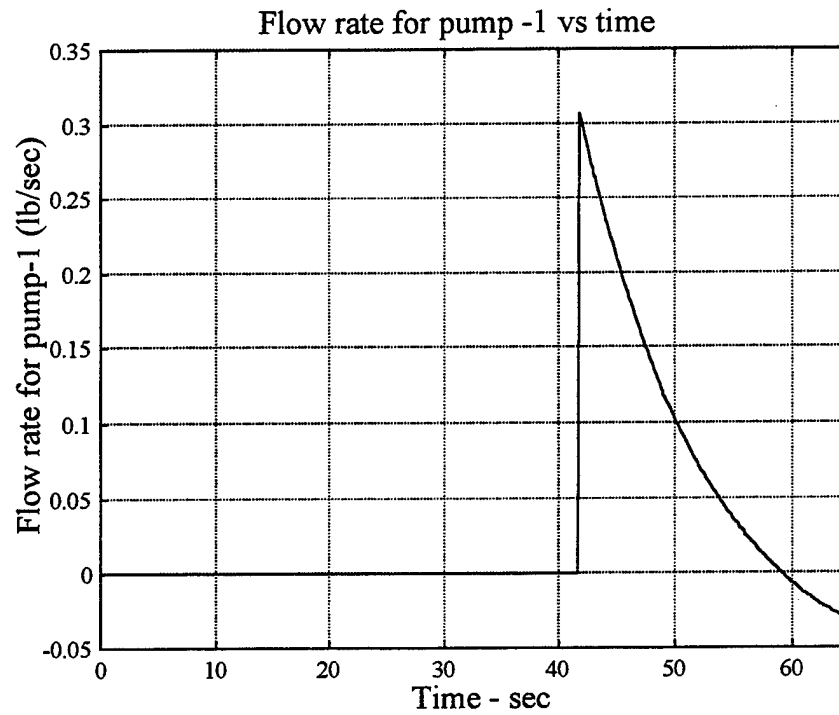
Figure 5-15. Flow Rate For Pump-1 With Weight And Depth Commands

Figure 5-16. Flow Rate For Pump-2 With Weight And Depth Commands
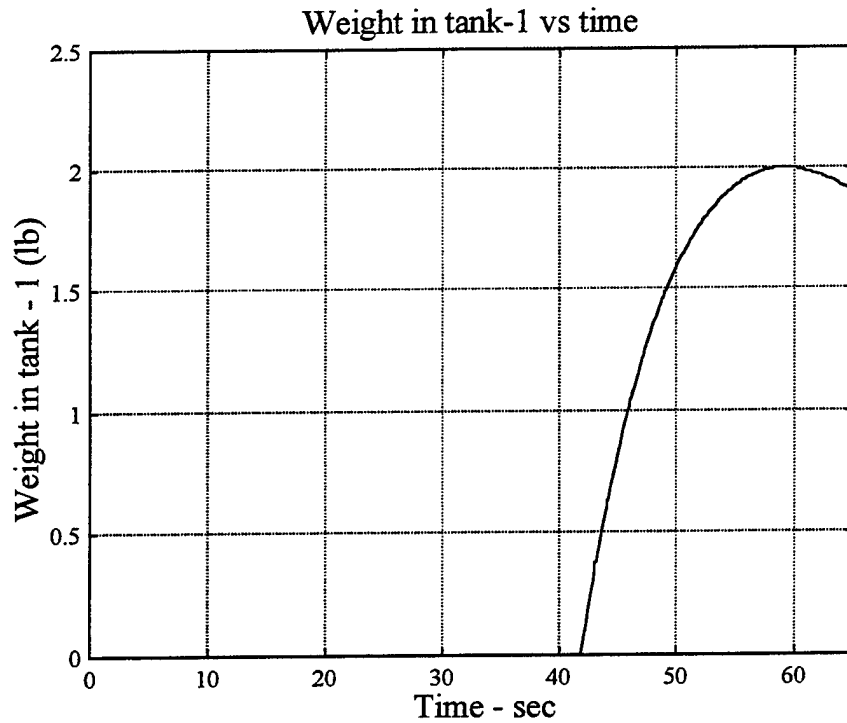
Figure 5-17. Weight Change In Ballast Tank-1 With Depth And Weight Commands
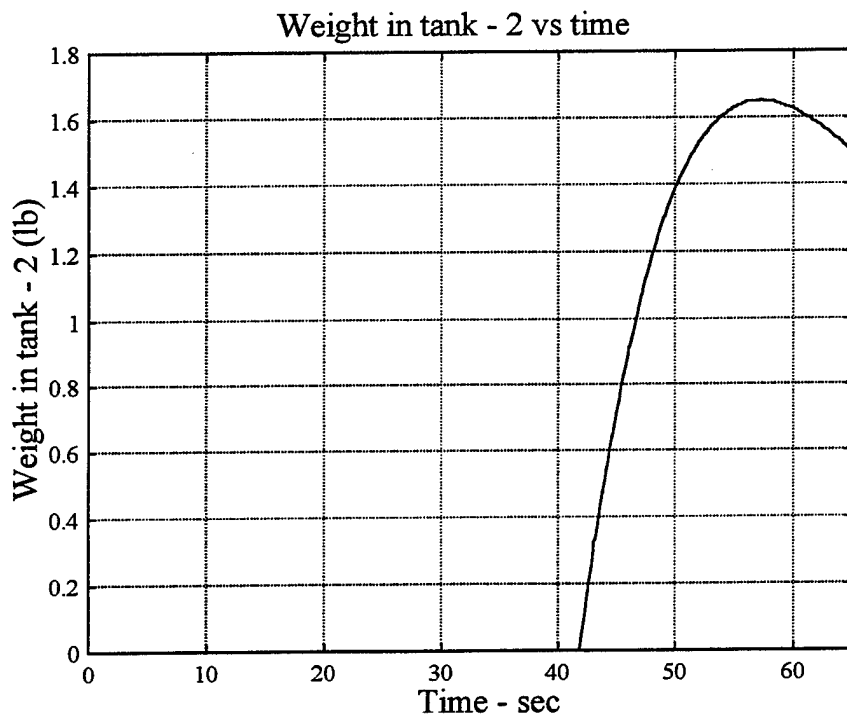
Figure 5-18. Weight Change In Ballast Tank-2 With Depth And Weight Commands

50

Depth vs time
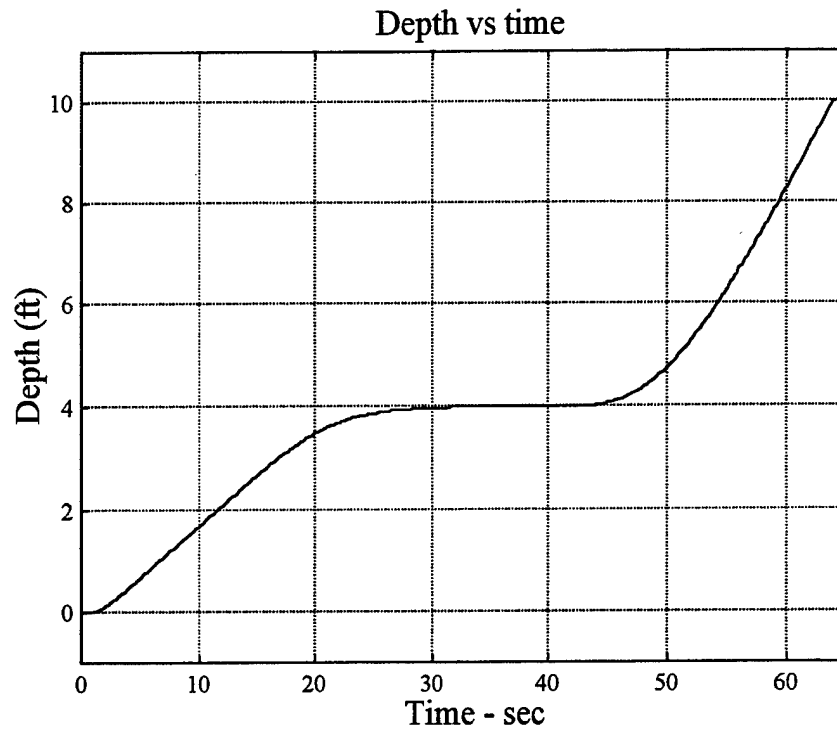
Figure 5-19. Depth Change With Weight And Depth Commands
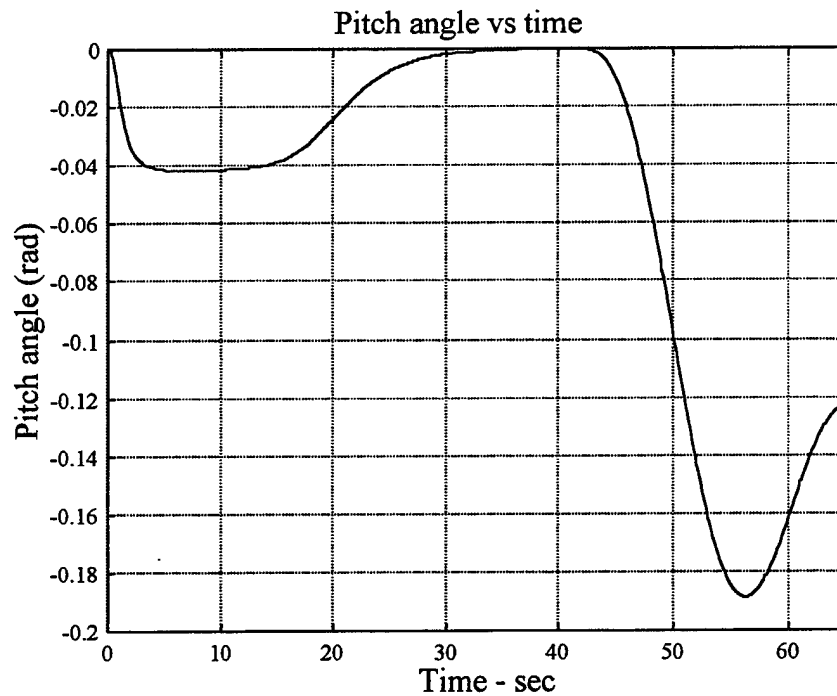
Pitch angle vs time

Figure 5-20. Pitch Angle As A Function Of Time With Weight And Depth Commands

51

F.   GROUNDING WITH THRUSTERS IN ADDITION TO WEIGHT CONTROL

NPS Phoenix AUV's cross-body thrusters consist of a 3 in. ID aluminum tube with a centrally located 4 blade brass propeller. A spur gear is mounted around a 3 in. diameter propeller and driven by a pinion connected to a 24 Vdc motor giving a 2.5:1 gear reduction. The twist of the propeller blade is symmetric enabling bi-directional operation delivering approximately 1.0 pound of bollard pull force in either direction [Ref. 11]. The system was simulated with the new state equations described in Chapter III and the same weight control conditions defined in section C of this chapter. Figure 5-21 through Figure 5-26 show the result of this simulation. The depth rate becomes 0.5 ft/sec and the pitch angle reaches a higher value since there is no control on thrusters. Because the thrusters are dominant in this case, the weight controller losses most of its effect on pitch control.

G.   BOTTOM STABILITY

The ground also affects the vehicle closing to the bottom. The theory explained by Hoerner and Borst [Ref. 12], predicts that roughly below $C_L =1.5$ (C_L represents lift coefficient), lift will be increased in proximity of the ground. Eventhough no experimental data is provided for NPS Phoenix AUV, it can be assumed that lift coefficient will be less than 1.5. For study of the bottom stability, two different cases of grounding were considered. Figure 5-27 shows these two cases.

In the case that the vehicle's stern touched to the bottom first, the lift will decrease the weight and inrease the angle of attack. This reduces the stability and makes grounding more difficult. But this feature can be very helpful when leaving the ground. In the other case, the bow of the vehicle touches the ground first. This time lift makes the vehicle heavier and decreases the angle of attack providing more stable grounding. After the completion of grounding process, the vehicle sits on the bottom with no lift since the lift coefficient, C_L is assumed zero because of the symmetric shape of the NPS Phoenix AUV. Figure 5-28 shows the ocean current that the vehicle

can stand with different ballast weights sitting on soil with 0.7 friction coefficient. When both ballast tanks are filled with water completely (~23.4 lb water in each tank), the vehicle can keep its position against 1.52 m/sec (~3 knots) of current.
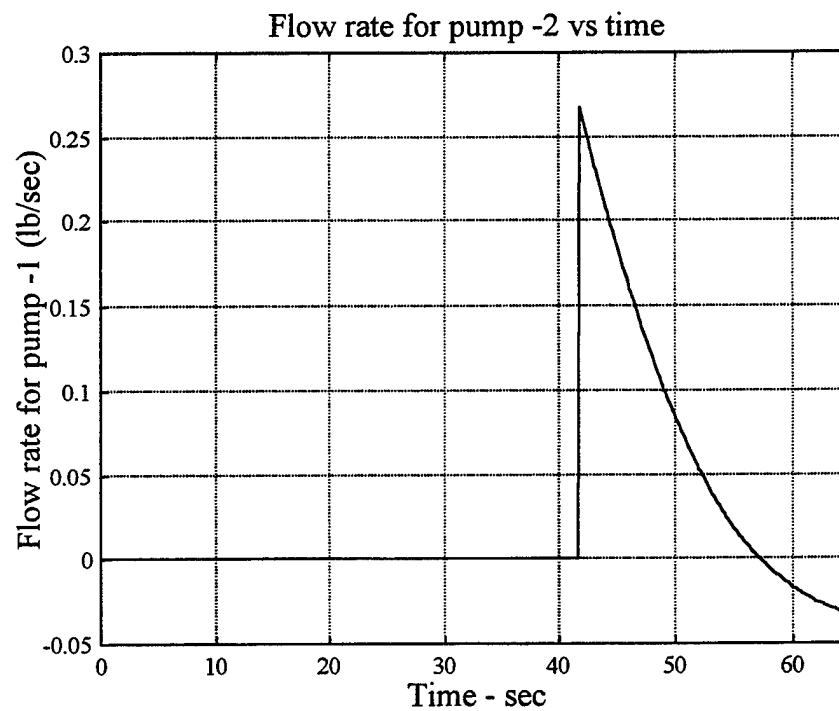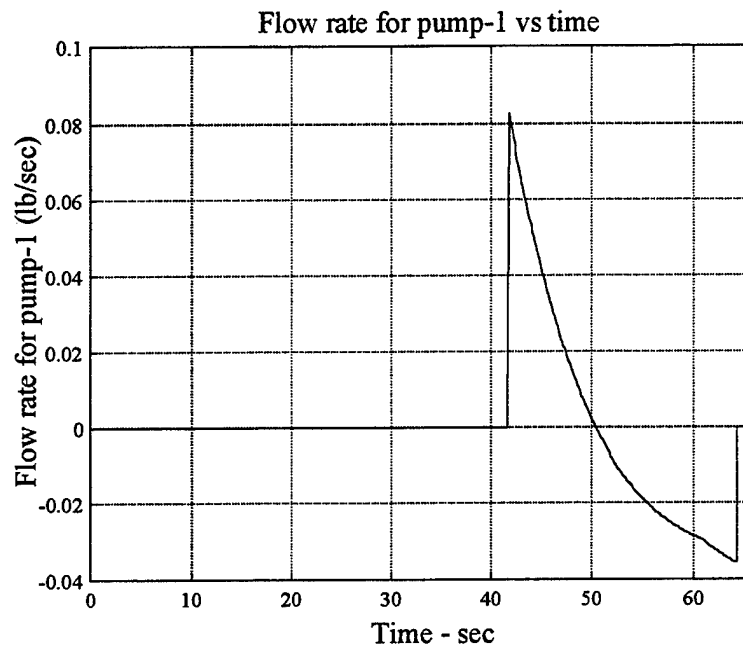

Figure 5-21. Flow Rate For Pump-1 With Thrusters And Weight Control
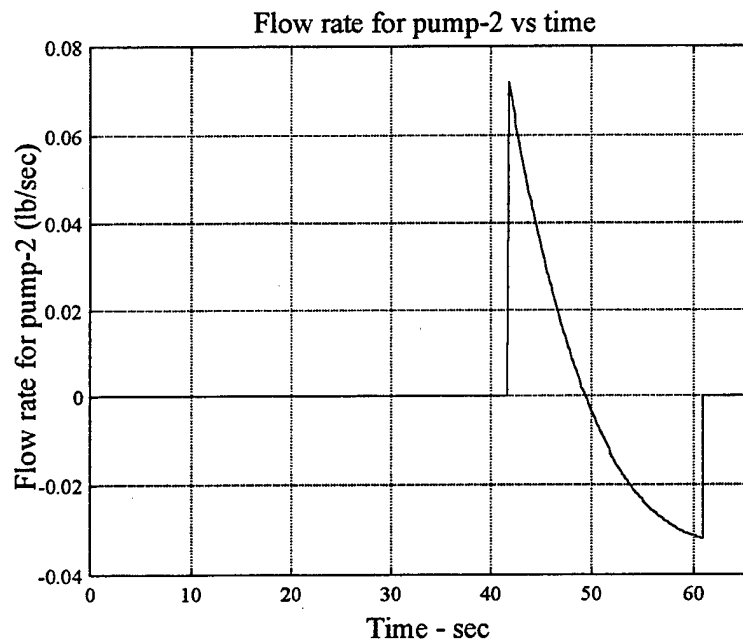
Figure 5-22. Flow Rate For Pump-2 With Thrusters And Weight Control

53
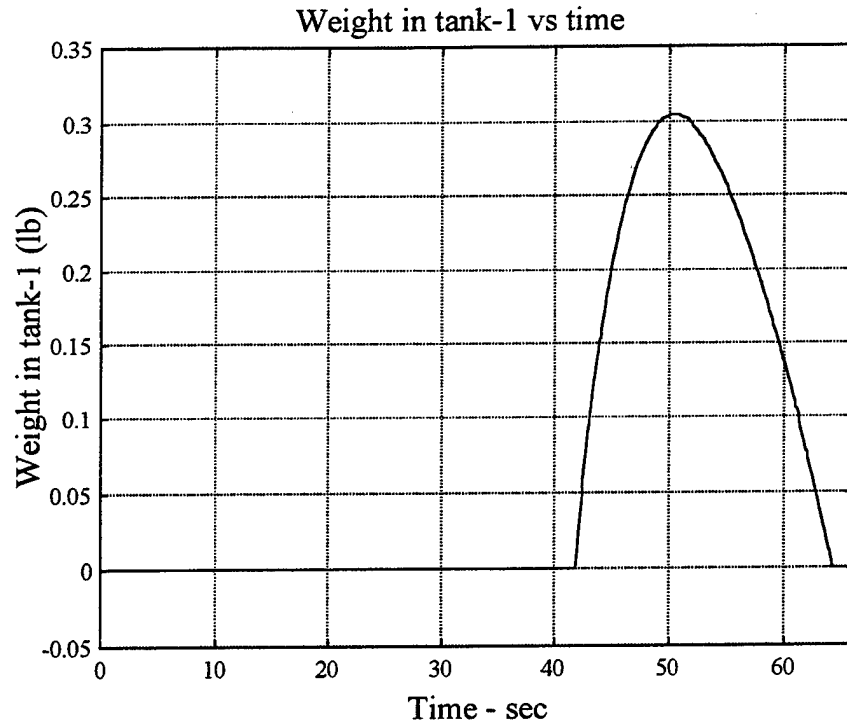
Figure 5-23. Weight Change In Ballast Tank-1 With Thrusters And Weight Control
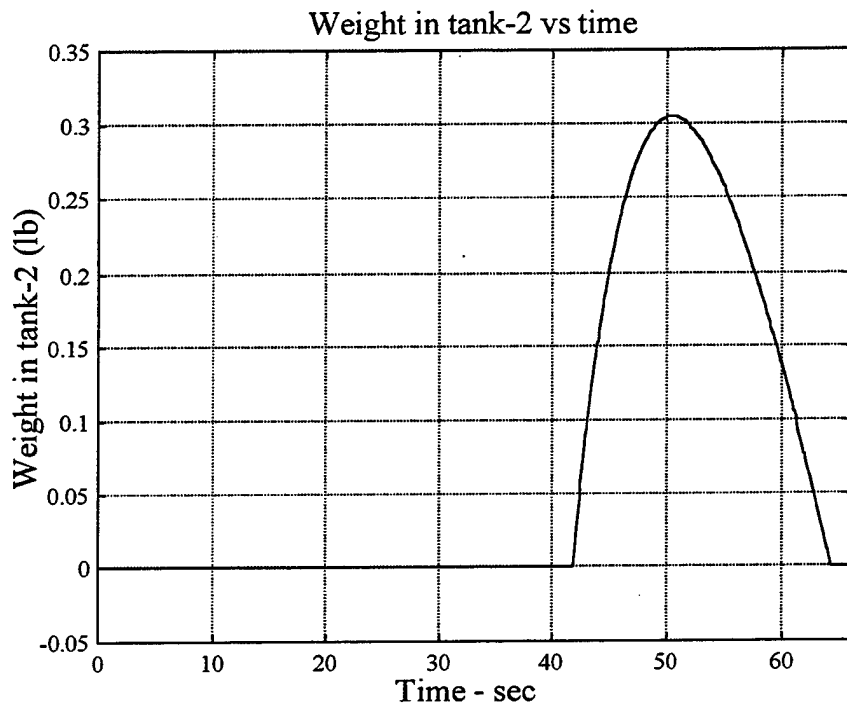
Figure 5-24. Weight Change In Ballast Tank-2 With Thrusters And Weight Control

54
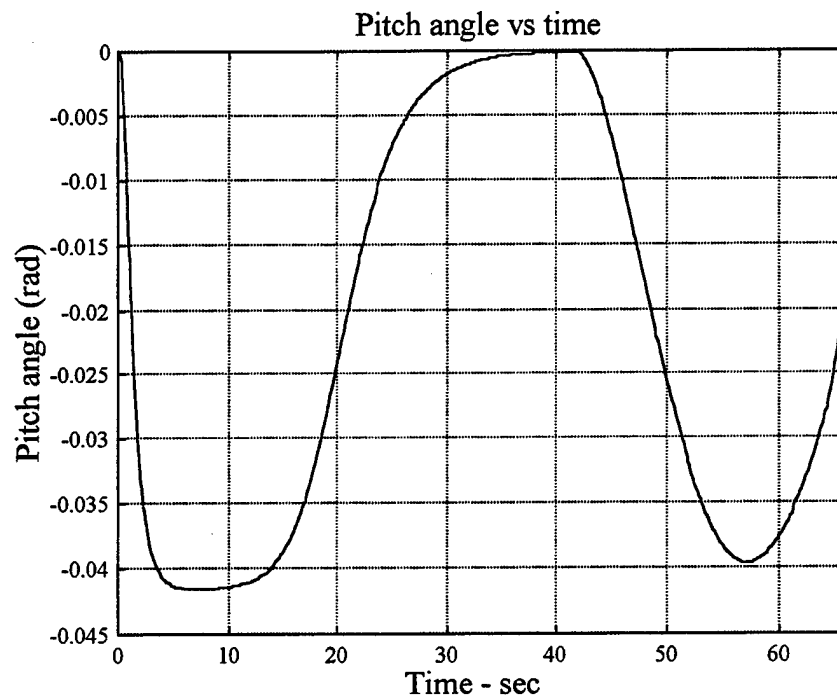
Figure 5-25. Pitch Angle As A Function Of Time With Thrusters And Weight Control
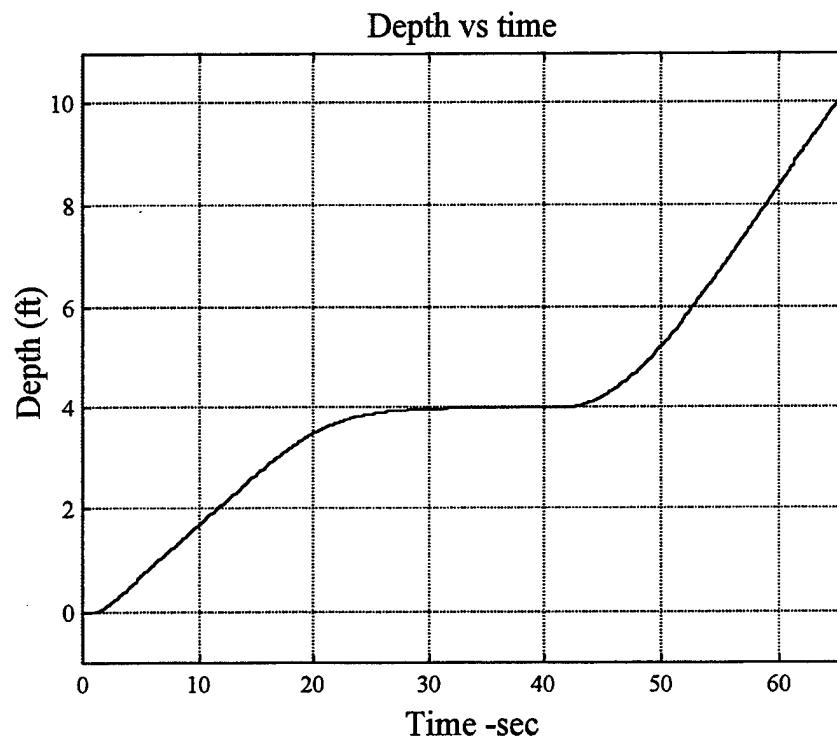
Figure 5-26. Depth Change During Flight And Grounding With Thrusters And Wt. Cont.

55
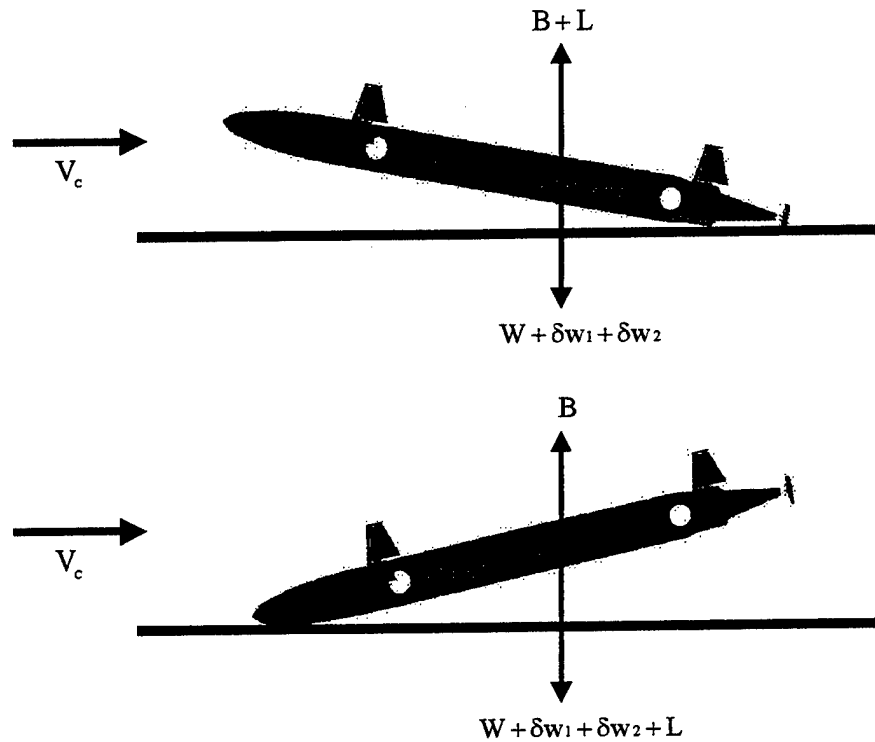
Figure 5-27. Forces Acting On The Vehicle In Two Cases Of Grounding
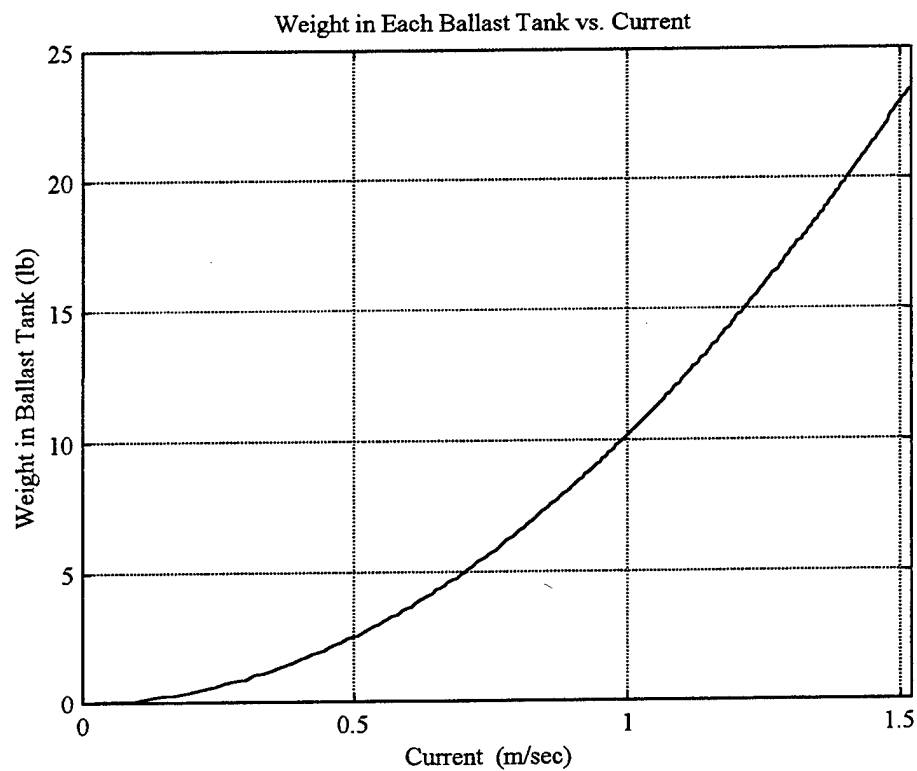
Figure 5-28. The Current That Vehicle Can Keep Its Position

VI.   PROPOSED HARDWARE IMPLEMENTATION

For NPS Phoenix Vehicle, a full description of all sensors and other hardware components which were present on the vehicle were given by Marco [Ref. 11]. It covers the gyroscopes, speed sensor, short baseline navigation system and GPS components. These components are used to sense vehicle roll, pitch and heading angular positions, depth, forward speed and location in global coordinates. Two open screws located at the stern, two vertical and two lateral cross-body thrusters, eight control surfaces (four rudders and four vertical planes) control these states. In the new design with dynamic ballast control, two tanks (bow and stern), two water pumps, and two flow meters as sensors were added to the vehicle. Those tanks and pumps were already shown on Figure 3-1. More detailed design can be seen on Figure 6-1.
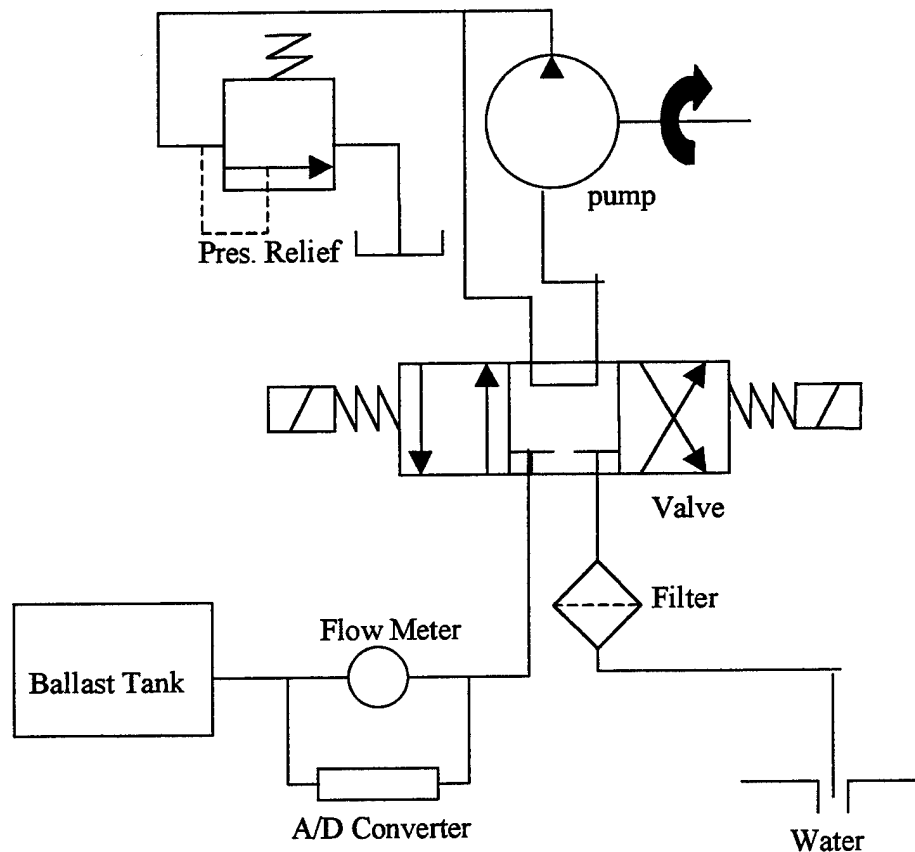
Figure 6-1. Hardware Components For Ballast Control System

The pump provides one way continuous water delivery with its positive displacement, 3 chamber diaphragm. Its maximum capacity is 1.75 gpm with 12 nominal dc voltage. Since the pump works only one way, a four connection, three position, spring centered, selonoid-control valve is used to change the flow direction. By simply changing the position of the valve, the same pump can be used to pump water both in and out of the ballast tank.

Flow meters will be used as sensors to provide flow rate information to the control system. Flow rate will be calculated from the pressure difference between the two ends of the meter and this analog data will be converted to digital one in A/D converter in order to be used in the main computer of the vehicle.

VII. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

• A method for modeling and simulation of a soft grounding system for an underwater vehicle has been developed and presented. This study outlined three different sections of that system, flight control, weight control and bottom stability.

• The critical parameters are pitch angle, depth rate and pump flow rate. The limitations of these parameters relative to the new NPS AUV have been determined and a feasible design has been validated by simulation.

• An LQR state feedback control provided sufficient weight keeping to maintain the vehicle pitch angle within the limits, but the depth change rate is limited. For a higher depth rate, the vertical thrusters should be used in addition to the weight control. This uses more energy, however, which may be unacceptable in practice.

• In ocean currents, the bottom stability depends on vehicle's direction and angle of attack. For stable grounding, the vehicle would commanded to face the opposite direction of the current. Bottom conditions are highly variable, but an estimate indicates that stability for currents less than 1 knot may be possible.

B. RECOMMENDATIONS

The most important recommendation and the next step should be to perform experiments in Monterey Bay with added weight in the vehicle. With the Sontek ADV velocimeter to measure water velocity, these experiments will determine weight requirements for bottom stability in typical ocean conditions.

APPENDIX A

The matlab programs used to simulate the dynamic behavior of the NPS Phoenix are given in this appendix

MATLAB PROGRAMS

```
%=================================================================
% deneme1.m is the main function of the dynamic system. It defines the parameters, variables and
% state vectors
%=================================================================
function [sys,x0,str,ts] = deneme1(t,x,u,flag)
global rho Zqdot Zwdot Mqdot Mwdot Wo g Iyo L


% The parameters used for the simulation
% The physical parameters of the vehicle
L   = 94.0/12.0;
L1  = 24.25/12.0;
L2  = 14.75/12.0;
Wo  = 435.0;
g   = 32.174;
Bu  = Wo;
Iyo = 42.0;
rho = 1.94;
xG  = 0.0;
xB  = xG;
ZG  = 0.5/12.0;
ZB  = 0.0;
ZGB = ZG - ZB;


% The hydrodynamic coefficients
Zqdot = -0.00253*2756.81;
Zwdot = -0.09340*377.67;
Zq    = -0.07013*377.67;
Zw    = -0.78440*51.72;
```

Zds = -0.02110*51.72;

Zdb = -0.02110*51.72;

Mqdot = -0.00625*20137.50;

Mwdot = -0.00253*2756.81;

Mq   = -0.01530*2756.81;

Mw   = 0.05122*377.67;

Mds = -1.7664;

Mdb = 1.3260;

% The bow and stern plane relations

a = 1;

ZGB = ZG - ZB;

Zd = (Zds + a*Zdb);

Md = (Mds + a*Mdb);

% A and B matrices were used to pass values to the other functions of dynamic system.

A=[0 0 1 0 0 0 0;Bu xG xB ZB 0 0 0;1 Zw Zq 1 0 0 0; ZG Mw Mq 1 0 0 0;1 1 0 0 0 0 0;

      0 0 0 0 0 0 0;0 0 0 0 0 0 0];

B=[0 0 0;1 1 Zd;L1 L2 Md;0 0 0;0 0 0;0 0 0;0 0 0];

% Since the system is full state feedback, C is a diagonal matrix

C=[1 0 0 0 0 0 0;0 1 0 0 0 0 0;0 0 1 0 0 0 0;0 0 0 1 0 0 0;0 0 0 0 1 0 0;0 0 0 0 0 1 0;0 0 0 0 0 0 1];

D=zeros(7,5);

switch flag,

% Dispatch the flag. The switch function controls the calls to S-function routines at each simulation stage

% Initialization

% sys is a generic return argument ; x0 is the initial state values ; str is provided only for consistency % with the S-function API for block diagrams ; ts is a two column matrix containing the sample times % and offsets of states associated with the block

case 0

62

```matlab
[sys,x0,str,ts] = mdlInitializeSizes(A,B,C,D);


% Calculate derivatives
% t is the time ; x is the state vector ; u is the input vector
case 1
 sys = mdlDerivatives(t,x,u,A,B,C,D);


% Calculate outputs
case 3
sys = mdlOutputs(t,x,u,A,B,C,D);


% Nothing is defined for calculation of next sample hit, updating discrete states and ending the
% simulation tasks
case {2,4,9}
sys = [];


otherwise
error(['Unhandled flag =',num2str(flag)]);
end
%  end deneme1


%==========================================================================
% mdlInitializeSizes
% Return the sizes, initial conditions, and sample times for the S-function
%==========================================================================
function [sys,x0,str,ts] = mdlInitializeSizes(A,B,C,D)


% Call simsizes for a sizes structure, fill it in, and convert it to a sizes array
sizes = simsizes;


% Number of continuous states
sizes.NumContStates = 7;
```

```matlab
% Number of dicrete states
sizes.NumDiscStates = 0;

% Number of outputs
sizes.NumOutputs   = 7;

% Number of inputs
sizes.NumInputs    = 5;

% Flag for direct feedthrough
sizes.DirFeedthrough= 0;

% Number of sample times
sizes.NumSampleTimes= 1;

% The following statement passes the information in the sizes structure to sys, a vector that holds
the % information for use by simulink
sys = simsizes(sizes);

% Initialize the initial conditions
x0  = [0 0 0 0 0 0 0];

% str is an empty matrix
str = [];

% Because the system is continuous, ts and its offset becomes 0.
ts  = [0 0];
%  end mdlInitializeSizes

%=================================================================
% mdlDerivatives
% Return the sizes, initial conditions, and sample times for the S-function
%=================================================================
```

```
function sys = mdlDerivatives(t,x,u,A,B,C,D)
global rho Zqdot Zwdot Mqdot Mwdot Wo g Iyo L


% Plane angle
dr = u(3);


% Current
Ucz = u(5);


% Velocity
v = u(4);


% State Variables
theta = x(1);
w = x(2);
q = x(3);
z = x(4);
Xs = x(5);
d1 = x(6);
d2 = x(7);


Bu=A(2); ZG=A(4); xG=A(9); Zw=A(10);
Mw=A(11); xB=A(16); Zq=A(17); Mq=A(18); ZB=A(23);
Zd=B(16); Md=B(17);
L1=B(3); L2=B(10);


%===============================================
% This part will be used only when the system is simulated with thrusters.
Zheave = 0;Mthrust = 0;
if abs(u(1)) > 0;
        Zheave = 2;Mthrust = 38.63;
end;
%===============================================
```

```
% Weight, mass , and moment of inertia
W = Wo + d1 + d2;
m = W/g;
Iy = Iyo + d1/g*(L1^2) + d2/g*(L2^2);


% M Matrix
Mo = [1 0 0 0 0 0 0;0 (m-Zwdot) (-m*xG-Zqdot) 0 0 0 0;0 (-m*xG-Mwdot) (Iy-Mqdot) 0 0 0 0;
        0 0 0 1 0 0 0;0 0 0 0 1 0 0;0 0 0 0 0 1 0;0 0 0 0 0 0 1];
% ==============================================
% This function calculates drag force and drag moment
 [Mdrag,Zdrag] = Mdrag(x);
% ==============================================
% Equations of motion v represents forw.spd. since u is used for pump flowrate
f1 = q;
f2 = (m+Zq)*v*q + m*ZG*(q^2) + Zw*v*w + (W-Bu)*cos(theta) +(v^2)*Zd*dr - Zdrag + Zheave;
f3 = (Mq-m*xG)*v*q - m*ZG*w*q + Mw*v*w - (xG*Wo-xB*Bu)*cos(theta) ...
    - (ZG*W-ZB*Bu)*sin(theta) + (v^2)*Md*dr +(-L1*d1 + L2*d2)*cos(theta) - Mdrag-Mthrust*q;
f4 = -v*sin(theta) + w*cos(theta);
f5 = v*cos(theta) + w*sin(theta) + Ucz;
f6=u(1);
f7=u(2);


sys = inv(Mo)*[f1;f2;f3;f4;f5;f6;f7];
%  end mdlDerivatives


%=====================================================================
% mdlOutputs
% Return the derivatives for the continuous states
%=====================================================================
function sys = mdlOutputs(t,x,u,A,B,C,D)
```

```
sys = C*x + D*u;
%  end mdlOutputs


%==================================================================
% Mdrag
% Makes calculations for drag terms in the equations of motion
%==================================================================
function [Mdrag,Zdrag] = Mdrag(x);
global rho Zqdot Zwdot Mqdot Mwdot Wo g Iyo L
CD = 1.2;


% Beam and length definitions for trapaziodal rule
xL=[0.0,-43.9/12.0,-39.2/12.0,-35.2/12.0,-31.2/12.0,-27.2/12.0,-
0.0/12.0,0.0/12.0,10.0/12.0,26.8/12.0,
32.0/12.0,37.8/12.0,40.8/12.0,42.3/12.0,43.3/12.0,43.7/12.0];
x1 = xL + L/2;


bm=[0.0,16.5/12.0,16.5/12.0,16.5/12.0,16.5/12.0,16.5/12.0,16.5/12.0,16.5/12.0,16.5/12.0,16.5/
12.0,15.5/12.0,12.4/12.0,9.5/12.0,7.0/12.0,4.0/12.0,0.0/12.0];


X2 = x(2);
X3 = x(3);


if X2 == 0,
  X2 = 1e-5;
end;


if X3 == 0,
  X3 = 1e-5;
end;


Zval = bm.*((X2-xL.*X3).^3)./(abs(X2-xL.*X3));
Mval = bm.*(((X2-xL.*X3).^3).*xL)./(abs(X2-xL.*X3));
```

```
ZDragval=0;
MDragval=0;


% Trapaziodal integration
for n=1:length(xL)-1,
Zdragval=0.5*(Zval(n)+Zval(n+1))*(xL(n+1)-xL(n));
Mdragval=0.5*(Mval(n)+Mval(n+1))*(xL(n+1)-xL(n));
ZDragval=ZDragval+Zdragval;
MDragval=MDragval+Mdragval;
end;


Zdrag=(0.5)*rho*CD*ZDragval;
Mdrag=(0.5)*rho*CD*MDragval;
% end Mdrag


%================================================================
% smcont
% Makes calculations for flight controller using sliding mode techniques
%================================================================
function plane_ang = smcont(u)
% State variable ;  theta, w, q, z
x = u(1:4);
% Commanded flight depth
zcom = u(5);
% Forward speed
v = 4;


% Matrices of linear equations of motion
Mo = [1 0 0 0;0 (m-Zwdot) (-m*xG-Zqdot) 0;0 (-m*xG-Mwdot) (Iy-Mqdot) 0;0 0 0 1];
Ao=[0 0 1 0;0 Zw*v (m+Zq)*v 0;-(ZG*W-ZB*Bu) Mw*v (Mq-m*xG)*v 0;-v 1 0 0];
Bo=[0;Zd*(v^2);Md*(v^2);0];


A = inv(Mo)*Ao;
```

68

```
B = inv(Mo)*Bo;

% Pole placement
pp = [-2,-2.1,-2.2,0];
kk = place(A,B,pp);
Ac = A-B*kk;

[mm,nn] = eig(Ac');
ss = mm(:,4);
xcom = [0;0;0;zcom];
sig = ss'*(x-xcom);

phi  = 0.1;
Nmax = 2;
eta = Nmax*0.4/inv((ss'*B));

% Control law
plane_ang = -kk*x-Nmax*0.4*sign(inv((ss'*B)))*tanh((sig/phi));

% end smcont
```

%===
% lq
% Calculates gains for weight controller using LQR techniques
%===

```
% Forward speed
v = 0.1;


% Matrices of linear equations of motion
Mo = [1 0 0 0 0 0;0 (m-Zwdot) (-m*xG-Zqdot) 0 0 0;0 (-m*xG-Mwdot) (Iy-Mqdot) 0 0 0;
        0 0 0 1 0 0;0 0 0 0 1 0;0 0 0 0 0 1];
Ao=[0,0,1,0,0,0;0,Zw*v,(m+Zq)*v,0,1,1;-(ZG*W-ZB*Bu),Mw*v,(Mq-m*xG)*v,0,-L1,L2;
```

69

```
     -v,1,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0];
Bo=[0,0;0,0;0,0;0,0;1,0;0,1];

A = inv(Mo)*Ao;
B = inv(Mo)*Bo;

Q = eye(6);
Q(1)=10000;
Q(22)=100;

R = 300000*eye(2);
[k,s,m]=lqr(A,B,Q,R);
k
% end lqr
```

```
%=================================================
% weightcont
% Calculates the flow rates for each ballast pump
%=================================================
function wcur = weightcont(u)

% state variables
x= u(1:6);

% forward speed
v = u(7);

% ground depth
zgr = 10;

xc = [0 0 0 zgr 0 0]';
xerr = x - xc;
```

% Matrices for linear equations of motion

Mo = [1 0 0 0 0 0;0 (m-Zwdot) (-m*xG-Zqdot) 0 0 0;0 (-m*xG-Mwdot) (Iy-Mqdot) 0 0 0;
      0 0 0 1 0 0;0 0 0 0 1 0;0 0 0 0 0 1];

Ao = [0,0,1,0,0,0;0,Zw*v,(m+Zq)*v,0,1,1;-(ZG*W-ZB*Bu),Mw*v,(Mq-m*xG)*v,0,-L1,L2;
      -v,1,0,0,0,0;0,0,0,0,0,0;0,0,0,0,0,0];

Bo=[0,0;0,0;0,0;0,0;1,0;0,1];

A = inv(Mo)*Ao;
B = inv(Mo)*Bo;

% Checks if the grounding procedure started
if v > 0.1;
   wcur = [0;0];
else
     % Gains found in lq function
   K1 = [0.0608   0.1263   -0.0526   0.0138   0.0754   0.0365];
   K2 = [-0.0322   0.1294   0.0021   0.0120   0.0365   0.0613];

% Flow rate for pump-1
wcur(1) = -K1*xerr;
% If there is no water in ballast tank-1 and flow rate for pump-1 is negative, then turns off the pump
if (u(5) <= 0 & wcur(1) < 0);
   wcur(1) = 0;
end;

% Flow rate for pump-2
wcur(2) = -K2*xerr;
% If there is no water in ballast tank-2 and flow rate for pump-2 is negative, then turns off the pump

71

```matlab
if (u(6) <= 0 & wcur(2) < 0);
  wcur(2) = 0;
end;
wcur = [wcur(1);wcur(2)];
end;
% end weightcont


%=======================================================
% speedcont
% Controls the forward speed
%=======================================================
function vcom = speedcont(u)


% Current velocity
vcur = u(1);
% Current depth
zcurrent = u(2);
% Commanded depth of flight
zcom = u(3);
z = zcom - zcurrent;
% Current longitudinal location
xreal = u(5);


% Commanded longitudinal location
xcomm = u(6);


% Bottom depth
zgrnd = u(7);
location = xcomm - xreal;
% deceleration
acc = 0.625;
% Forward speed during the flight
Vo = u(4);
```

```
% Time
t = Vo/acc;
xstop = Vo*t-0.5*acc*(t^2);
zvert = zgrnd - zcom;
tvert = zvert/0.11;
xstop2 = 0.6*tvert + 9;
% Distance needed to complete stop
xstop1 = xstop + xstop2;


% Checks if the vehicle is in the commanded depth and reduces to speed
if abs(z) < 0.05;
   if location < xstop1;
        vcen = abs(u(1) - 0.1) ;
        if vcen > 0.1;
                       vcom = vcen;
                else
                       vcom = 0.1;
     end;
   else
        vcom = u(4);
   end;
else
        vcom = u(4);
end;
% end speedcont



%================================================
% seccont
% Keeps the vehicle's velocity at 0.1 ft/sec during grounding
%================================================
function vsec = seccont(u)
% Difference between bottom and current depth
```

```matlab
grnd = u(4) - u(5);

if (u(1) > 0 | u(3) > 0);
    vsec = 0.1;
elseif grnd > 3;
    vsec = 0.1;
else
    vsec = u(2);
end;
% end seccont


%=========================================================
% planeangcont
% Keeps the plane angles at 0 degree during grounding
%=========================================================
function ang = planeangcont(u)
if u(1) > 0.1;
    ang = u(2);
else
    ang = 0;
end;
% end planeangcont
%=========================================================
% grounddepth
% Stops the simulation at the ground depth defined by the user
%=========================================================
function grnd = grounddepth(u)
% Bootom depth
ground = u(1);
% Current depth
actdpth = u(2);

if actdpth > ground;
```

```
    grnd = ground;
else
   grnd = actdpth;
end;
% end grounddepth
```

APPENDIX B

The parameter values used to simulate the dynamic behavior of the NPS Phoenix are given in this appendix :

A. PHYSICAL PARAMETERS

W	Vehicle Weight	= 435 lbs
B	Vehicle Buoyancy	= 435 lbs
L	Characteristic Length	= 94 in.
L_1	Bow Ballast Tank Offset from C.G.	= 24.25 in.
L_2	Stern Ballast Tank Offset from C.G.	= 14.75 in.
I_{yz}	Moment of Inertia	= 42 lb-ft-sec^2
x_G	x Coordinate of C.G. From Body-Fixed Origin	= 0.0 in
x_B	x Coordinate of C.B. From Body-Fixed Origin	= 0.0 in
z_B	z Coordinate of C.B. From Body-Fixed Origin	= 0.0 in
z_G	z Coordinate of C.G. From Body-Fixed Origin	= 0.5 in

B. CONTROL INPUTS

δ_b	Bow Plane Deflection
δ_s	Stern Plane Deflection
f_1	Bow Ballast Pump Flow Rate
f_2	Stern Ballast Pump Flow Rate

C. NON-DIMENSIONALIZED HYDRODYNAMIC COEFFICIENTS

$Z'_{\dot{q}} = -0.00253$	$Z'_{\dot{w}} = -0.09340$	$Z'_q = -0.07013$
$Z'_w = -0.78440$	$Z'_{\delta_s} = -0.02110$	$Z'_{\delta_b} = -0.02110$
$M'_{\dot{q}} = -0.00625$	$M'_{\dot{w}} = -0.00253$	$M'_q = -0.01530$
$M'_w = 0.05122$	$M'_{\delta_s} = -1.7664$	$M'_{\delta_b} = 1.3260$

LIST OF REFERENCES

1. Schenkenberger, J., "Jens Schenkenberger's Luftschiff Homepage." [http://people. Frankfurt. netsurf.de/Jens.Schenkenberger/]. 1998

2. Motyka, P., and Bergmann, E., "The Design of a Control System for the Ballast and Trim of an Unmanned Submersible", *Proceedings of the 1984 American Control Conference,* San Diego, California, v.3, pp 1786-1793, 6 June 1984

3. DeBitetto, P.A., "Fuzzy Logic for Depth Control of Unmanned Undersea Vehicles", *IEEE Journal of Oceanic Engineering*, v.20, no.3, pp 242-247, July 1995

4. Healey, A.J., *Dynamics and Control of Mobile Robotic Vehicles,* Lecture Notes, chp. 2, pp 3-19, 1995

5. Riedel, J.S., *Pitchfork Bifurcations and Dive Plane Reversal of Submarines at Low Speeds,* Engineer's Thesis, Naval Postgraduate School, Monterey, California, June 1993

6. Papoulias, F.A., *Dynamics of Marine Vehicles,* Lecture Notes, chp. 4, pp 48-61, 1993

7. Hawkinson, T.D., *Multiple Input Sliding Mode Control for Autonomous Diving and Steering of Underwater Vehicles,* Master's Thesis, Naval Postgraduate School, Monterey, California, December 1990

8. Papoulias, F.A., *Modern Control Systems,* Lecture Notes, chp. 6, pp 95-101, 1992

9. Whitcomb, L.L., and Yoerger, D.R., "Comparative Experiments in the Dynamics and Model-Based Control of Marine Thrusters", *Proceedings of the 1995 IEEE Oceans Conference,* San Diego, California, v.1, pp 1019-1028, October 1995

10. Yoerger, D.R., Cooke, J.G., and Slotine, J.E., "The Influence of Thruster Dynamics on Underwater Vehicle Behavior and Their Incorporation Into Control System Design", *IEEE Journal of Oceanic Engineering,* v. 15, no. 3, pp 167-178, July 1990

11.   Marco, D.B., *Autonomous Control of Underwater Vehicles and Local Area Maneuvering*, Ph.D. Dissertation, Naval Posgraduate School, Monterey, California, September 1996

12.   Hoerner, S.F., and Borst, H.V., *Fluid Dynamic Lift*, chp. 20-21, Mrs. Liselotte A. Hoerner, 1975

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center..2
   8725 John J. Kingman Road, Ste 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library..2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Dr. A. J. Healey, Code ME/Hy...2
   Department of Mechanical Engineering
   Naval Postgraduate School
   Monterey, CA 93943

4. Dr. Thomas Curtin (Code 32OE)..1
   Office of Naval Research (ONR)
   800 North Quincy Street
   Arlington, VA 22217

5. Dr. Samuel M.Smith..1
   Dept of Ocean Engineering
   Florida Atlantic University
   500 N.W. 20 Street
   Boca Ratan, FL 33431-0991

6. Dr. Jim Bellingham..1
   MIT Sea Grant Program
   MIT
   Cambridge, MA 02139

7. Dr. Antonio Pascoal...1
   Institute Superior Tecnico
   Av.Rovisco Pais,
   1096 Lisboa Codex, PORTUGAL

8. Richard Blidberg...1
   Autonomous Undersea Systems Institute
   86 Old Concord Turnpike
   Lee, NH 03824

9. LCDR Jeffery S. Riedel, USN..1
   80 Boot Pond Rd
   Plymouth, MA 02360